AFWL-TR-79-193

AD-E200607

LEVEL III

AFWL-TR-79-193

AD A092379

QUANTA - USERS MANUAL

1st Lt M. E. Badessa

July 1980

Final Report

Approved for public release; distribution unlimited.

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base, NM 87117

80 12 02 007

AFWL-TR-79-193

This final report was prepared by the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico, under Job Order 88092220. First Lieutenant Maria Badessa (NTYV) was the Laboratory Project Officer-in-Charge.

This technical report has been reviewed and is approved for publication.

Maria E. Badessa

MARIA E. BADESSA
1st Lt, USAF
Project Officer

FOR THE DIRECTOR

Terry A. Schmidt

TERRY A. SCHMIDT
Major, USAF
Chief, Environment & Effects Branch

Thomas W. Ciambrone

THOMAS W. CIAMBRONE
Colonel, USAF
Chief, Applied Physics Division

DISCLAIMER NOTICE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFWL-TR-79-193	2. GOVT ACCESSION NO. AD-A092 379	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) QUANTA - USERS MANUAL,		5. TYPE OF REPORT & PERIOD COVERED Final Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 1st Lt Maria E. Badessa		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Weapons Laboratory (NTYV) Kirtland Air Force Base, NM 87117		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62601F/88092220
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Weapons Laboratory (NTYV) Kirtland Air Force Base, NM 87117		12. REPORT DATE July 1980 13. NUMBER OF PAGES 178
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Survivability
Base Escape
Weapon Allocation
Flush
Optimization
Nuclear

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report provides instructions for using the QUANTA computer code, a code developed within the Applied Physics Division of the Air Force Weapons Laboratory to study the survivability of a flushing aircraft force from a nuclear attack of sea launched ballistic missiles. Included topics are structure of the input deck, description of the restart capabilities, details of array sizes, and a list of the error messages that might be encountered. The most current version of the QUANTA code is attached as Appendix A.

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SUMMARY

The basic purpose of this report is to assist the QUANTA user in running the code efficiently. Included topics are structure of the input deck, description of the restart capabilities, details of array sizes, and a list of the error messages that may be encountered. It should be expected that occasional problems with the program will occur as the user attempts runs with new combinations of parameter values. If a problem is detected which cannot be solved through this documentation, notify the author or the Air Force Weapons Laboratory. Appendix A is a listing of the current version of QUANTA and Appendix B is the output of a run using the sample input data described in the text.

CONTENTS

INTRODUCTION

QUANTA is a computer model developed at the Air Force Weapons Laboratory (AFWL) to study the effects of a sea-launched ballistic missile (SLBM) nuclear attack on targets consisting of a flushing aircraft force. The program QUANTO (Ref. 1) was used as a basis for its mathematical development. Using the technique of Lagrangian multiplier optimization, a near-optimal allocation of SLBMs to targets is calculated. There are three main differences between these two codes. The first and most dramatic difference is that QUANTA simulates the base escape survivability in greater detail. QUANTO essentially attacks the circle about the turn point (not brake release point) where dispersed aircraft may possibly be at the time of weapon arrival. If aircraft are still on the runway or sitting at the brake release point, they still are considered to be evenly distributed within this circle. However, QUANTA attacks the target in two distinct positions: the runway and the circle centered at the turn point, thus giving a more accurate description of the probability of survival for flushing aircraft. The second difference is that QUANTA allows the attacker the option to allocate weapons to the nonalert aircraft at a base by assigning a value to that target. The third is that QUANTA does not allocate weapons which do not attain a certain value of aircraft killed. This value is user specified and will be discussed later in greater detail in the description of input.

This document contains the information needed to construct the input data for QUANTA and make a successful run. In addition, procedures for running with a restart capability are discussed. Notes on array dimensions and problem sizes are also included.

---

1. QUANTO--A CODE TO OPTIMIZE WEAPON ALLOCATIONS, AFWL-TR-73-242, Kirtland Air Force Base, New Mex., 1974.

INPUT DATA

The input is read by the subroutines STARTR, READER, PROCES, MARVIN, and ALOCIN in that order. A brief description of the variables that are read and their format will follow. The structure of the data deck is divided into the following categories:

- Initial data card
- Beddown data
- Aircraft parameters
- Submarine data
- Missile parameters and profile
- Aircraft profile
- Cutoff values
- Convergence parameters
- Initial allocation

STARTR will read the initial data card. Variables which determine the sizes of the arrays in QUANTA are on this card.

- Initial data card

  Format: 8I5

  Variables:

  NTGTS - number of targets/air bases

  NSUBS - number of submarine locations

  NTYPES - number of types of aircraft

  MXRWAY - maximum number of runways found on a given base

  MTYPES - number of types of missiles

  IOUT - If it equals 1, standard printout. Additional reallocation information will be printed, if it equals 2.

  NACTT - total number of aircraft in scenario

  NMARV - number of cutoff values. If all weapons in the scenario are to be allocated, NMARV must equal 0.

READER will read the beddown data, aircraft parameters, submarine data, and missile parameters and profile. The aircraft beddown data is input as a set of cards for each base. The first card(s) of the set identifies the base, with the number of alert aircraft at that target and is followed by a takeoff sequence list for the alert aircraft. The number of these sets must equal the value of

NTGTS, read by STARTR. The aircraft parameters such as hardness and brake release time will be read from one card per aircraft type. However, one parameter, the distance to the turn point, will be input for all aircraft types preceding aircraft data. The time intervals for takeoff sequences consist of a set of cards containing NTYPES variables. Submarine data will be read from one card per submarine location. The total number of cards must equal NSUBS (which was read from initial data card). The missile parameters and profile is input as a set of cards for each missile type. On the first card of the set will be the missile parameters such as yield and reliability. Following this card will be the missile profile.

- Beddown data

  Format: 3F10.4, I5, 5X, 3F10.4/(7F10.4)

  Variables:

  TGTLAT - target latitude in decimal degrees

  TGTLNG - target longitude in degrees

  BASVAL - value of the base which also could include a value for nonalert aircraft

  NRWAYS - number of runways on base

  VALj - number of aircraft of type j on base, j=1 to NTYPES

  Format: 70I1

  Variables:

  ISEQ - takeoff sequence of aircraft identified by type number

- Aircraft parameters

  Format: 7F10.6

  Variables:

  DTCENTj - the minimum distance from brake release (start of takeoff roll) to the turn point of aircraft type j, j=1 to NTYPES.

  Format: 4F10.4

  Variables:

  RELVAL - relative value of aircraft

  BRTIME - the time, in minutes, between launch of the first SLBMs and the start of takeoff roll of the aircraft. (Assume it is the first to take off.)

  PSI - overpressure hardness in lb/in^2

  CAL - vulnerability level to free-field thermal energy in cal/cm^2

Format: 7F10.4

Variables:

$DELTAC_{i,j,k}$ - takeoff time intervals in minutes between each ordered pair of aircraft. Type k will follow type j. After all possible combinations are read for single runways, then input dual, triple, etc. i=1 to MXRWAY, j=1 to NTYPES, k=1 to NTYPES. The last integer increments most often; e.g., (1,1,1), (1,1,2), ..., (1,1,NTYPES), (1,2,1), ..., (1,2,NTYPES), etc.

- Submarine data

Format: 2F10.4, 3I5

Variables:

SUBLAT - submarine latitude in degrees

SUBLNG - submarine longitude in degrees

ISUBS - number of submarines at this location

NMPS - number of missiles per submarine. (This number will automatically be the same for each sub at this location.)

MTYPE - missile type by number

- Missile parameters and profile

Format: 7F10.4

Variables:

DELTM - launch interval between salvos in minutes

REL - overall reliability of the missile. This includes launch, in-flight, and detonation.

RNGMIN - minimum range in nautical miles

RNGMAX - maximum range in nautical miles

YIELD - warhead yield in kilotons

BURST - height of burst in feet aboveground

RVS - number of reentry vehicles per missile

Format: I5, 5X, 6F10.4/(6F10.4)

Variables:

MPROF - number of data points in missile profile

FMTIME, FMRNG - time in minutes, distance in nautical miles of missile trajectory

PROCES will read the aircraft profile. A set of cards for each aircraft type consists of the following:

- Altitude
- Count

- Flight profile data - measured relative to the brake release for a nonturning aircraft.
- Additional parameters - needed to construct an extended flight profile in ground range and time.
- Environments - must be the same for all types of aircraft.

- Aircraft profile
  - Altitude

    Format: F10.8

    Variables:

    AZ - final level-off altitude in feet above the ground. It must be less than or equal to the maximum (last) altitude in the flight profile.

  - Count

    Format: I5

    Variables:

    NDATA - number of data points in aircraft profile

  - Flight profile data

    Format: 3F15.8, 2F10.6

    Variables:

    F - ground range in feet

    G - time in seconds

    A - altitude in feet which is nondecreasing

    VEL - Mach number describing ground speed

    ACCEL - level-off acceleration in feet per second squared if the aircraft were to level-off at the altitude specified on this card.

NOTE: Accurate values for distance, time and altitude are important because of the method QUANTA uses to compute kill regions for an aircraft. At the time of weapon arrival, the altitude and distance from brake release of the aircraft are used to calculate the probability of kill. However, the values for VEL and ACCEL are used to accelerate the aircraft from its velocity at level-off altitude to its final escape velocity. The standard equations of motion use these parameters associated with level-off only to determine time-distance pairs for the flight profile. Therefore, it would suffice if every number in VEL and ACCEL arrays were equal to the velocity and acceleration of the aircraft at level-off altitude.

  - Additional parameters

    Format: 3F15.8, F10.6

    Variables:

    ALTFM - minimum altitude in feet at which the aircraft attains final Mach

FMACH - final (maximum) Mach number

TINT - time interval, in seconds, used to extend the profile after the aircraft has reached its final escape velocity (60 s is presently used).

XATI - number of data points (in time and distance) that are calculated for the flight profile from the level-off velocity to final escape velocity (10 is presently used).

NOTE: The aircraft is assumed to follow the input flight profile until the level-off altitude is reached. QUANTA then accelerates the aircraft to its final escape velocity which is assumed to be the maximum velocity and is specified as FMACH. However, this cannot occur if XATI is equal to zero or if ALTFM is less than the level-off altitude. If XATI equals zero (i.e., there are no data points between level-off and escape velocity), no acceleration takes place after level-off and the velocity at level-off will define the final escape velocity. If ALTFM is less than the level-off altitude, then there is no acceleration since the aircraft should have reached its final mach in the input profile. After the aircraft has reached final escape, TINT and FMACH are used to determine time-distance pairs. The total number of data points in the extended profile is NXAP.

- Environments

Format: 4F10.8

Variables:

HTE - terrain height

RHO - the albedo factor, which is the fraction (between 0.0 and 1.0) of incident thermal energy which reflects off the ground. For a SAC normal day, the albedo is 0.3. A worst case abnormal day has a higher albedo.

VIS - the visibility in miles; 10 for a SAC normal day

PZ - the water vapor in the air, in millimeters; 5 for a SAC normal day

MARVIN will read the cutoff values. SLBMs which arrive at a target sufficiently late may not produce adequate kill to justify allocation. These SLBMs may be the last (or next to the last) weapons launched. This is a result of the aircraft having enough time to disperse into a larger area. The program will check if a weapon can attain a kill at least as large as the cutoff value. If it does not, that weapon will not be allocated to any real targets but to a dummy target. After an optimum allocation has been reached using this cutoff value, the program will use it as an initial allocation for the next run with a different cutoff value. For this reason, the cutoff values should be in descending order; i.e., highest value read first. This enables the program to be run more efficiently saving the user cost and time. If the cutoff is as low as 0.1,

almost all weapons will be allocated. If it is near 10.0, as little as one-tenth of the weapons will be allocated depending on the scenario.

- Cutoff values

  Format: 7F10.5

  Variables:

  CUT_i - the slope of the curve defining the relationship between the number of weapons allocated and the probability of kill, i=1 to NMARV.

ALOCIN will read the convergence parameters and the initial allocation. There are three methods by which the interation process will stop: (1) if the increase in kill becomes insignificant, (2) if the limit for the total number of iterations has been reached, or (3) if the difference in the Lagrangian multipliers becomes relatively small. These criteria are user specified. The initial allocation of missiles to targets is input submarine by submarine. The subs are in the order of the submarine locations with all subs at one location preceding the first sub at the next location.

- Convergence parameters

  Format: F10.4, 2I5, F10.4

  Variables:

  CHGKIL - the value of aircraft killed used as a measure for a significant increase in kill in ITCUT2 iterations, i.e., if in the last ITCUT2 iterations the value killed has not increased by at least CHGKIL, the iterative process will stop.

  ITCUT1 - the total number of ITCUT2 iterations in the optimization process, i.e., QUANTA, if not stopped by some other means, will continue through ITCUT1 x ITCUT2 number of iterations.

  ITCUT2 - after this number of iterations, QUANTA checks for a significant increase in value killed.

  EPSCUT - for an optimum, the Lagrangian multipliers must all be equal. Since this is impossible, a test for convergence is if they are within a tolerance of EPSCUT.

- Initial allocation

  Format: 14I5

  Variables:

  ITGINO - the target number to which the SLBM is allocated. The SLBMs from a submarine will appear in ascending order by salvo. If there are more than 14 salvos on a sub, then continue the allocation on the next card. Begin reading the allocation of the next sub with a new card. If the cutoff value is large, the initial allocation should read - 1 which represents the dummy target.

SAMPLE INPUT DATA

To assist the reader in developing an input deck, the following pages detail an example of the input data for QUANTA. The first row of numbers on each page of Table 1 are to aid the reader in locating the columns for the variables, while the first number on every line indicates the card number. The first card of input tells that there are four targets, three sub locations, two aircraft types, a maximum of two runways, two missile types, standard printout, 20 aircraft, and one marval run.

The next eight cards identify the four targets. The first base, located somewhere in the Pacific, has a value of 15 for eit its importance as a target or for nonalert aircraft. This is to insure that weapons will be drawn down. Aircraft type one precedes aircraft type two in the takeoff sequence on every base. The only target that has two runways is target three.

The following cards describe the two aircraft types. According to the tenth card, the first aircraft has the capability to make its flyout turn 3.5 nmi from brake release. The following two cards identify the aircraft types with their relative value, brake-release time, and overpressure and thermal hardnesses. According to cards 13 and 14, aircraft type one following type one has a take-off interval of 0.25 min. If dual runways exist, the time interval is cut in half.

The following groun of nine cards describes submarine and missile parameters. Cards 15 through 17 state the three submarine locations along with the number of missiles on each sub at that location. There are three subs at location two, while locations one and three have only two subs. These cards also show a total of 36 missiles of type one and 56 missiles of type two in this scenario. However, there are 40 weapon groups (NWPNS described later). A weapon group consists of weapons that have the same characteristics, i.e., they are launched at the same time, from the same location, and have the same parameters. The next six cards identify missile types one and two with weapon reliability, yield, height-of-burst and number of RVs. A total of four data points are used to describe the trajectory for each missile.

The next group of 20 cards details the flight profile for aircraft type one. Card number 24 says that this aircraft reaches level-off altitude of 10,000 ft., but the flight profile indicates that it reaches a maximum Mach of 0.9 at a

TABLE 1. SAMPLE DATA DECK

```
    1234567890123456789012345678901234567890123456789012345678901234567890
1.      4    3    2    2    2    1   20    1
2.    13.5833  144.9167   15.0        1       2.0       1.0
3.  112
4.    44.8      68.8333    2.0        1       1.0       3.0
5.  1222
6.    33.9167   84.5167               2       4.0       5.0
7.  111122222
8.    33.4333  112.1667   10.0        1       4.0
9.  1111
10.    3.5       5.5
11.    2.5       3.5       1.0      96.0
12.    1.0       4.2       1.5     120.0
13.    0.25      0.5       0.5       0.5      0.125     0.25       0.25
14.    0.25
15.   36.75     74.0        2   18    1
16.   28.05     93.45       3   12    2
17.   45.65    126.05       2   10    2
18.    0.2       0.9     250.0    2250.0    1000.0    7000.0       2.0
19.     4        3.5     250.0       5.5     950.0       7.5    1600.0
20.    9.5    2250.0
21.    0.25      0.85    650.0    2000.0     800.0    6000.0       3.0
22.     4        5.0     650.0       7.5    1100.0      10.0    1550.0
23.   12.5    2000.0
24. 10000.0
25.    16      PROFILE 1
26.              0.0            0.0            0.0       0.0
27.           1500.0            7.5            5.0      0.13
28.           3700.0           19.0            7.0      0.17
29.           5700.0           30.0           16.0      0.20
30.           9000.0           48.0           75.0      0.25
31.          13000.0           63.0          230.0      0.34
32.          20000.0           79.5          250.0      0.45
33.          33000.0          105.0          500.0      0.60
```

TABLE 1. CONTINUED.

```
     1234567890123456789012345678901234567890123456789012345678901234567890
34.          50000.0          130.0         1250.0      0.75
35.          63000.0          150.0         2300.0      0.80
36.          72000.0          160.0         2800.0      0.90
37.          90000.0          180.0         4500.0      0.90
38.          98000.0          195.0         5800.0      0.90
39.         105000.0          200.0         7000.0      0.90
40.         110000.0          205.0         9200.0      0.90
41.         115000.0          210.0        10000.0      0.90
42.  2800.0        0.9               60.0
43.     0.0        0.3     10.0       5.0
44.  5000.0
45.     11        PROFILE 2
46.               0.0           0.0            0.0                 6.2
47.           2000.0           20.0            9.0      0.14       6.2
48.           3900.0           30.0           30.0      0.14       6.2
49.           6200.0           40.0           92.0      0.19       6.2
50.           9500.0           50.0          360.0      0.25       6.2
51.          13400.0           60.0          500.0      0.31       6.2
52.          15600.0           66.0         500.01      0.35       6.2
53.          55900.0          125.0         500.11      0.68       6.2
54.          61000.0          132.0         1425.0      0.70       6.2
55.          68000.0          140.0         3950.0      0.72       6.2
56.          75500.0          148.0         5000.0      0.73       6.2
57.  7600.0         0.849          60.0            10.0
58.     0.0        0.3     10.0       5.0
59.     0.5
60.     0.1        10  100  0.01
61.      1     2    3    4   1    2    3   -1   -1   -1   -1   -1   -1   -1
62.     -1    -1   -1   -1
63.      1     2    3    4   4    2    3    1   -1   -1   -1   -1   -1   -1
64.     -1    -1   -1   -1
65.      1     1    1    2   2    2    3    3    3    4    4    4
66.      1     1    1    2   2    3    2    3    3    4    4    4
```

TABLE 1. CONCLUDED.

1234567890123456789012345678901234567890123456789012345678901234567890

67.	4	4	4	3	3	3	2	2	2	1	1	1
68.	1	0	2	0	3	0	4	0	-1	-1		
69.	1	0	2	0	3	0	4	0	-1	-1		

lower altitude of 2,800 ft. This is stated on cards 36 and 42. For this reason, the acceleration is zero on all profile cards and on card 42 XATI is also zero. QUANTA will simply extend the profile using FMACH and TINT (card 42) to find time-distance pairs. Cards 44 through 58 describe the flight profile for aircraft type two. The first card of this group shows a level-off altitude of 5,000 ft. After it reaches this altitude it will accelerate at 6.2 ft/s^2 (card number 56) until it reaches a final mach of 0.849 at an altitude of 7,600 ft (card number 57). A total of 10 data points will identify this interval in the flight profile (XATI on card 57). At 60 s intervals, distance will be computed by QUANTA to complete the profile. Please note that cards 43 and 58 have the same values for the environments.

For this one marval run, the cutoff value on card 59 is 0.5. The next card, which is the convergence parameters card, states the three conditions which may terminate the iterative process: (1) after every 100 iterations, QUANTA will check for an increase of kill of at least 0.1 value points, (2) there will be no more than 1,000 iterations, and (3) if the difference between the largest and smallest Lagrangian multiplier is less than 0.01, an optimum solution has been reached.

The last group of cards which describes the initial allocation matrix consists of a total of nine cards. The first seven missiles from each sub at location one will be allocated to real targets (cards 61 and 63). The rest of the weapons will go to the dummy target since the user feels they will not attain enough kill. All of the missiles at sub location two will be allocated while some at the third sub location will not. This is shown by the zeros appearing on the last two cards. This is a mathematical representation of missiles being used for other time urgent targets such as communications.

RESTART OPTION IN QUANTA

The restart option, pertinent for CDC systems, permits the QUANTA user to resume calculations for a particular scenario without the loss of an excessive amount of time. For example, if the computer should crash after the program has been executing for 20 minutes, all is not lost. With a restart tape, the user can recover the computations close to the point of interruption.

There are three methods by which QUANTA writes a restart tape. First, if sense switch one is turned on (controlled by the operator), the program will write the essential variables onto the restart file. In the example mentioned above, the operator would turn the switch on before the computer crashed, thereby enabling QUANTA to write the tape and stop execution gracefully. The second method will also terminate the problem for insufficient time. After ITCUT2 iterations have been completed in the optimization process, QUANTA verifies that there is ample time for another ITCUT2 iteration. Third, during each run, the restart file will be automatically written after every fifth ITCUT1 iteration (i.e., 5 x ITCUT2, 10 x ITCUT2, etc.). This is to insure that the user can still obtain a tape even if something disastrous should happen. If there is any doubt the program will not be completed in the time allotted, the user should save the restart tape.

If the user chooses to save the file, it must be specified in the control language during the initial run. The local file name of the tape is TAPE13 and should be referenced as such throughout the initial run and all subsequent runs. The QUANTA user has the option of storing the restart file on magnetic tape or disk. When starting with a restart tape that is on permanent file, sense switch two must be turned on. If the sense switch is on and the file does not exist, QUANTA will abort. A sequence of commands appearing in the control cards for obtaining a permanent file on an initial run is:

```
REQUEST (TAPE13, *PF)
COPY (ZILCH, TAPE13)
FTN (I = COMPILE, S, S = PFMTEXT, . . .)
  .
  .
  .
CATALOG (TAPE13, FILENAME, ID = . . .)
```

and for that restart run,

```
ATTACH (TAPE13, FILENAME, ID = . . .)
SWITCH(2)
FTN (I = COMPILE, S, S = PFMTEXT, . . .)
   .
   .
   .
CATALOG (TAPE13, FILENAME, ID = . . .)
```

The first two cards in the above restart run replace the first two cards of the initial run. During all QUANTA runs, the FTN card requires the S, S = PFMTEXT parameters for compilation. The user may choose to catalogue TAPE13 on all subsequent runs.

At this point it becomes very apparent that the restart capability in QUANTA is machine dependent in its language. In particular, the ALTER and EXTEND routines are in job control language for the CDC computers. Before attempting to use the program on a computer other than the CDC, consult computer manuals for equivalent commands.

On a restart run, data are read just as in the initial run. QUANTA will use these data in the beginning of the program to set up arrays and compute the survivability matrix which for storage purposes is too large to be put onto the tape. When QUANTA begins the reallocation process (for finding the optimal laydown), it will check a restart file, if one exists. The information from that file will be used as a closer approximation to the optimal solution. Figure 1 demonstrates the above procedures.

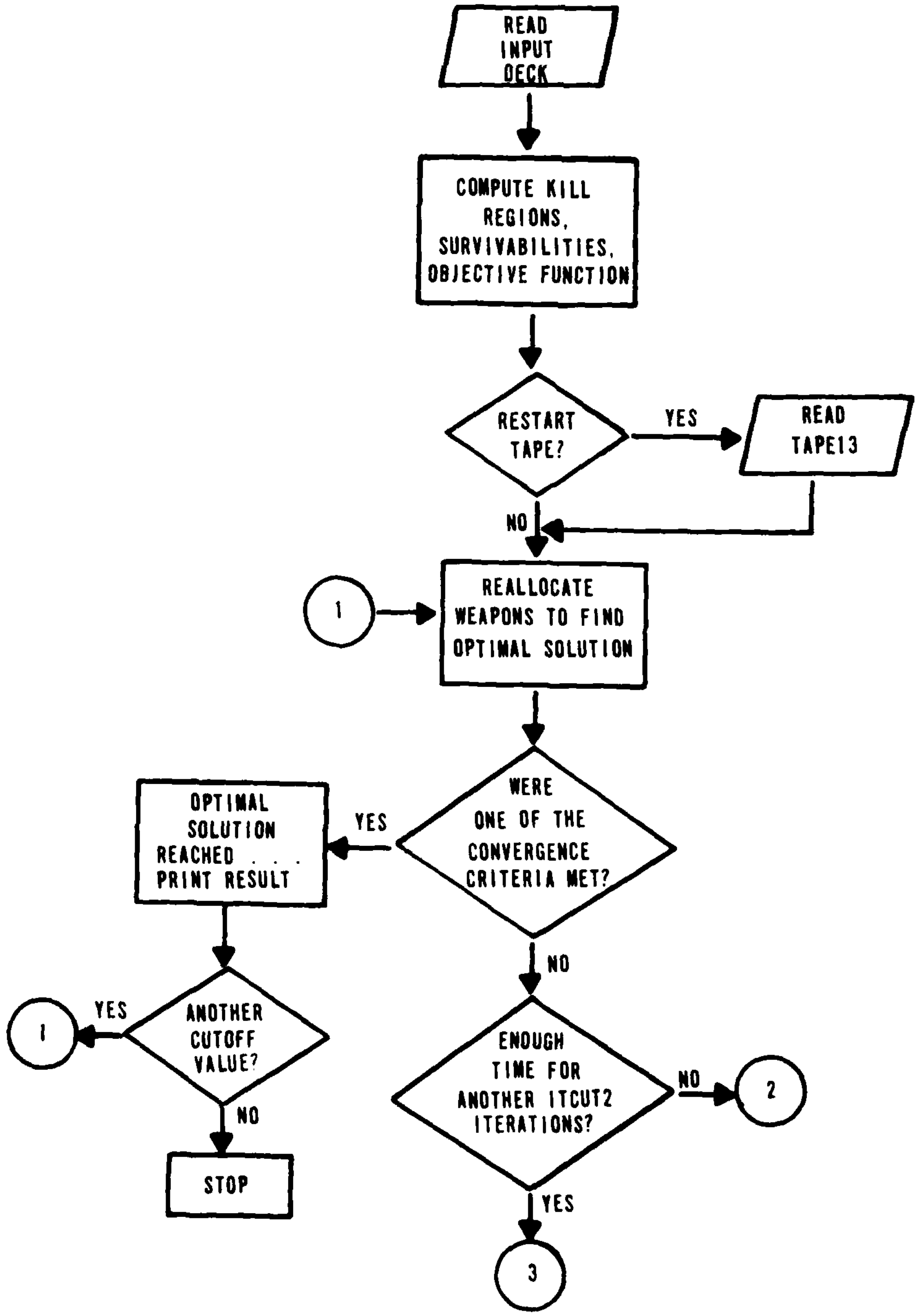

Figure 1. Flow chart.

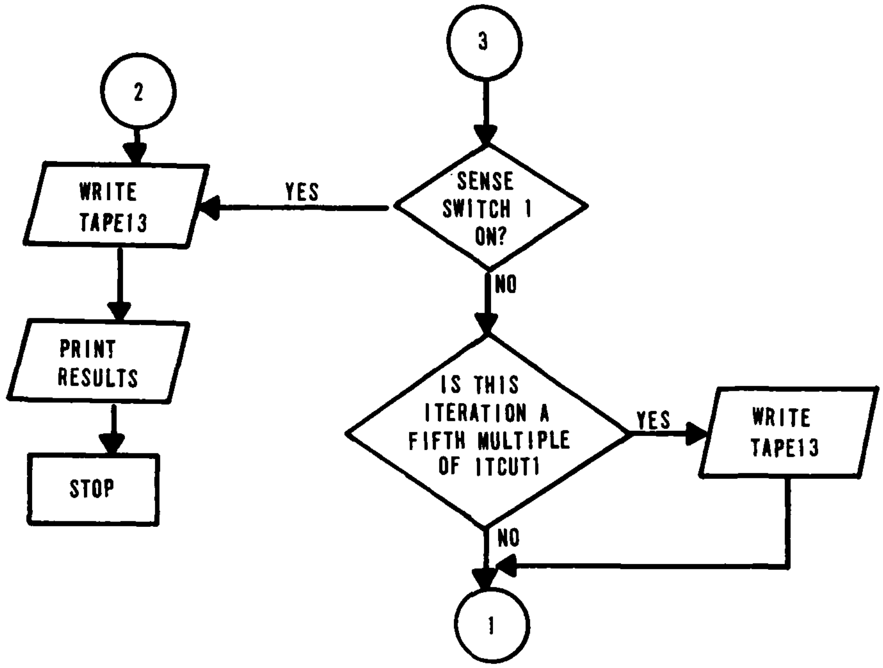

Figure 1. Concluded.

ARRAY DIMENSIONS

This section is to give the reader a better understanding of the program itself and to delineate the variables within the code that must be changed with problem size. Internally, QUANTA sets up four large arrays. In the main routine, they are named WORK1, WORK2, ALOC, and FLAMB.

The WORK1 array comprises a number of small arrays which have been read from input and will be utilized throughout the execution of the program. One example is the array containing the relative value of the aircraft. This variable is essential in the calculation of the total probability of kill which is continuously recomputed during the iterative process.* There are 12 such arrays in the WORK1 array and its size is determined by the subroutines WORK1 and REMW1. The formula for the size is given below.

After QUANTA reads and verifies all the input, it begins calculations for the survivability of aircraft at a target against all weapons. Since the size of this matrix can become extremely large, these computations are stored onto tape (TAPE4) for mass storage (machine dependent). After this is completed, most of the variables, such as target and submarine locations, needed for these calculations are no longer required. Therefore, in order to save core, the WORK2 array, originally storing such variables, is then used for computations which define the objective function. For this reason, two messages are printed onto output for the required WORK2 array size. The first message tells how much storage is needed for the input variables, while the second tells the amount of core needed to compute the probability of kill. The subroutines WORK2, REMW2, and SECW2 calculate the size and the formulas for both which are given below. The user should make the size of the WORK2 array at least as large as the maximum of the two.

The size is the same for both the ALOC and FLAMB arrays. The ALOC array stores the allocation matrix and the FLAMB array is used for the Lagrangian multiplier matrix. Their dimension should be set to 3 x NWPNS x NTGTS, where NTGTS is the number of targets and NWPNS is the number of weapon groups, not total number of weapons. To compute NWPNS add the number of missiles (described as NMPS in input) at each submarine location (not each sub at the location).

*Through this reallocation process, QUANTA finds an optimal missile laydown by moving weapons from target to target.

If the user has not provided adequate core for all these arrays, an error message will be printed and execution stopped. The program does this check by comparing the values of NLCM1, NLCM2, and NLCM3, with the required dimension sizes. A data statement in the main program defining NLCM1, NLCM2, and NLCM3 should be set to the number of words in WORK1, WORK2, and ALOC, respectively. Stated below are the variables which determine the dimensions of the arrays. Most of them are read from the initial card of input.

NLCM1 = NTGTS (number of targets)
+ (NTGTS + 1) x NTYPES (types of aircraft)
+ 3 x NSUBS (number of sub locations)
+ MTYPES (types of missiles)
+ NACT (number of aircraft)
+ NMARV (number of marval runs)
+ NWPNS (total number of weapon groups)
+ 2 x NWPNS x MXNAPB (maximum number of aircraft found at a given target)

NLCM2 = 3 x NTGTS
+ 5 x NTYPES
+ NTYPES x NTYPES x MXRWAY (maximum runways on a given target)
+ 63 x NTYPES x MTYPES
+ 7 x MTYPES
+ 2 x NSUBS (number of sub locations)
+ 2 x NTYPES x NXAP (number of data points in the extended aircraft profile)
+ 6 x NTYPES x NPAP (maximum of data points in the aircraft profile)
+ 2 x MTYPES x MXMPR (maximum points in missile trajectory)
+ 2 x MXNAPB
+ MXNMPS (maximum of the number of missiles per sub - NMPS)
+ 30

NLCM2* = NACT
+ 2 x NTGTS
+ 3 x NTYPES
+ MTYPES
+ 4 x MXNAPB

*The value defined in the program should be the maximum of the two NLCM2 formulas.

NLCM3 = 3 x NTGTS x NWPNS

There is a data statement in the main routine which may affect the size of WORK2. The variables NPAP, NXAP, and MXMPR are in the equation for the first NLCM2 above. NPAP is the maximum of the number of data points (described in input as NDATA) of all the aircraft profiles. NXAP is the desired number of time-distance pairs in the extended profile for all aircraft types. MXMPR is the maximum of the number of pairs (MPROF in input) in the missile trajectories. These values should be appropriately defined in the DATA statement.

There is another array in the main program that must change with problem size. INDX sets up the index for reading and writing the large survivability matrix on mass storage. Its dimension must be at least 2 x NTGTS + 1. A data statement defining NDX must also be set to this value. As with the WORK arrays, if the size is incorrect, an error message will be printed indicating the problem.

Outside of the main program, in particular in four subroutines, there are arrays which must be dimensioned properly to insure complete execution. In subroutines SETS, OUTRNG, and ZERO an array BASENO must be the size of NTGTS. The data statement which checks its correct size is in subroutine ZERO. In subroutine BOSS arrays FN, ST, and STT must be at least as large as NMARV (the number of marval runs). The data statement initializing these arrays must also be changed.

With each new problem, the user must insure that the dimensions of all these arrays are adequate for storage. With each array there is an integer that must equal the value of its dimension. The purpose is to allow QUANTA's built-in check system to verify sufficient core for problem size.

ERROR MESSAGES

In order to minimize the number of errors that could occur in such a large code as QUANTA, a series of checks were developed for the input data. However, these checks do not enable the program to sense when a typographical error has been made by the user. For example, it cannot determine that the aircraft overpressure hardness was meant to read 1.0 when in actuality it reads 10. For this reason, the program will print the variables as they are read so that the user may verify the quantities. QUANTA also checks the variables when read but in a general sense. For example, it can detect if the aircraft vulnerability levels and weapon yields are positive values.

Table 2 lists all the error messages QUANTA could issue and the remedies for these errors. When an error is detected, the program will stop execution immediately following a message printed in the day file and at the end of output. The latter message will identify the subroutine along with an error number and a brief description of the problem.

TABLE 2. ERROR DESCRIPTION

Subroutine/Error No.	Message	What To Do
REMW1		
1	Size of WORK1 array exceeds dimension.	The dimension of the WORK1 array is too small for your problem size. Change its dimension to what the required size reads. Also change NLCM1 in the data statement.
SECW2		
1	Size of WORK2 array exceeds dimension.	The dimension of the WORK2 array is too small for your problem size. Change its dimension to what the required size reads. Also change NLCM2 in the data statement.
REMW2		
1	Size of WORK2 array exceeds dimension.	Same as SECW2-1.
STARTR		
1-5	Variable read must be greater than 0.	Change the first card of input to read nonnegative numbers.
6	IOUT must be either 1 or 2.	Change the sixth variable on the first card of input to comply with message.
7-8	Variable read must be greater than 0.	Change NACT or NMARV to comply.
READER		
1	Total number of aircraft is greater than original input.	The number of aircraft that was read in is less than the total number added from each base. Check the number of aircraft at each base. If that is correct, then change the first card (NACT). Otherwise, change the number at the base which is incorrect.

TABLE 2. CONTINUED.

Subroutine/Error No.	Message	What To Do
READER (cont.)		
2	Base latitude and longitude must be greater than 0.	Change the variables so that they are positive.
3	Number of runways on a base must be greater than 0 and not greater than the maximum number of runways.	Check to make sure that all bases have the correct number of runways. If it is not right, change that particular card. Otherwise, change the first card of input (MXRWAY) to reflect the maximum of the number of runways on a base.
4	Number of aircraft of a given type cannot be less than 0.	Change the variable VAL so that it is nonnegative.
5	An aircraft type on a base must be greater than 0 and not greater than the number of types.	Change the takeoff sequence ISEQ so that the values read will be from 1 to NTYPES (which is read on the first card).
6-13	Variables read must be greater than 0.	Change the variables so that they are positive.
14	The type of sub must be greater than 0 and not greater than the total.	Change the missile type of a sub location so that its value is between 1 and MTYPES (which is read on the first card of input); or, change MTYPES to comply.
15	The number of time/distance pairs cannot be greater than MXMPR.	MXMPR is a data statement in the main program which is the maximum of the number of pairs in the missile profile. Check that number against the profiles of each type. Change the appropriate value.
16	The time interval between salvo launches must be greater than 0.	Change DELTM so that it is positive.

TABLE 2. CONTINUED.

Subroutine/Error No.	Message	What To Do
READER (cont.)		
17	The reliability of a missile must be greater than 0 and less than 1.	Change the value of REL to comply with message.
18	The maximum range of the missile must be greater than its minimum range.	Change the maximum range RNGMAX or the minimum range RNGMIN to comply.
19-23	The variables read must be greater than 0.	Change the variables to comply with messages.
PROCES		
1	The level-off altitude of the aircraft must be greater than 0.	Change AZ to a positive value.
2	The number of data points in aircraft profile is larger than NPAP.	NPAP is in a data statement in the main program. It is the maximum of the number of points in the aircraft profile. Check it against all aircraft types. Change the appropriate value.
3-5	Variables read cannot be less than 0.	Change the first card in the profile so that the variables are nonnegative.
6-7	The ground range and time must be strictly increasing.	Check that each variable read on consecutive cards is increasing in value. Change those that are not.
8-11	Variables read must be positive.	Change the additional parameters card so that the variables comply with messages.
12-15	Variables read must be positive.	Change the variables on the environments card so that they meet the requirements specified in the error message.

TABLE 2. CONCLUDED.

Subroutine/Error No.	Message	What To Do
PROCES (cont.)		
16	Mach at level-off cannot be greater than final Mach.	Check the Mach at the level-off altitude and FMACH to insure that the values comply with error message.
MARVIN		
1	CUT cannot be less than 0.	Change the variable so that it is positive.
ALOCIN		
1-4	Variables read must be greater than 0.	Change the convergence parameters card so that the variables are positive.
5	The target number *receiving* this weapon must be between -1 and NTGTS.	Change the variables to comply with error message.
DTLINE		
1-8	Root cannot be found.	Check the vulnerability levels for the aircraft and weapon yield for validity.

APPENDIX A

QUANTA LISTING

```
PAGE   1

      PROGRAM QUANTA(INPUT,OUTPUT,TAPE4,TAPE13,TAPE5=INPUT,TAPE6=OUTPUT) QUANTA   2
      COMMON/LCM1/WORK1(806)                                              QUANTA   3
      COMMON/LCM2/WORK2(656)                                              QUANTA   4
      COMMON/LCM3/ ALOC(408)                                              QUANTA   5
      COMMON/LCM4/ FLAMB(408)                                             QUANTA   6
      COMMON/TRM/ISSW1,IJOIT                                              QUANTA   7
      COMMON /RANIOX/ ISET,INDX(9)                                        QUANTA   8
      LEVEL 2,WORK1,WORK2,ALOC,FLAMB                                      QUANTA   9
      DATA NLCM1,NLCM2,NLCM3/806,656,480/                                 QUANTA  10
      DATA NPAP,NXAP,MAMPR/16,20,4/                                       QUANTA  11
      DATA ISET/0/, NDX/9/                                                QUANTA  12
C             CALL STARTR TO BEGIN THIS CASE                              QUANTA  13
      CALL STARTR(NTGTS,NSUBS,NTYPES,MARWAY,MTYPES,IOUT,NACT,NMARV)      QUANTA  14
      CALL ZERO (NTGTS,NSUBS)                                             QUANTA  15
      CALL OPENMS (4,INDX,NDX,0)                                          QUANTA  16
      NR=2*NTGTS +1                                                       QUANTA  17
      WRITE(6,601) NR,NDX                                                 QUANTA  18
      IF(NR.GT.NDX) STOP  " RANIOX  "                                     QUANTA  19
C             SET UP WORK ARRAYS                                          QUANTA  20
      CALL WORKR1(NTGTS,NTYPES,NSUBS,NVAL,NRELVA,NISUBS,NNMPS,NMTYPE,    QUANTA  21
     * NWORK1,NLCM1,NISEQ,NACT,NREL,MTYPES,NBASVA)                        QUANTA  22
      CALL WORKR2(NTGTS,NSUBS,NTYPES,MARWAY,MTYPES,NPAP,NXAP,MAMPR,      QUANTA  23
     * NLCM2,NTGTLA,NTGTLN,NDTCEN,NNRWAY,NBRTIM,NPSI,NICAL,NDELTA,      QUANTA  24
     * NSUBLA,NSUBLN,NDELTM,NRNGM1,NRNGMA,NYIELD,NBURST,NRVS,NMPROF,    QUANTA  25
     * NFMTIM,NFMRNG,NGPTS,NGAREA,NGRLMA,NS,ND,NA,NF,NG,NVS,NVEL,NACCEL, QUANTA  26
     * NNDATA,NFLRP,NFTSA,NFLRT,NWORK2)                                   QUANTA  27
C             READ AIRCRAFT AND SUBMARINE DATA                            QUANTA  28
      CALL READER(NTGTS,NSUBS,NTYPES,MTYPES,MARWAY,NACT,MAMPR,WORK1      QUANTA  29
     * (NVAL),WORK1(NRELVA),WORK1(NISUBS),WORK1(NNMPS),WORK1(NMTYPE),   QUANTA  30
     * WORK1(NISEQ),WORK2(NTGTLA),WORK2(NTGTLN),WORK2(NDTCEN),WORK2     QUANTA  31
     * (NNRWAY),WORK2(NBRTIM),WORK2(NPSI),WORK2(NICAL),WORK2(NDELTA),   QUANTA  32
     * WORK2(NSUBLA),WORK2(NSUBLN),WORK2(NDELTM),WORK1(NREL),WORK2      QUANTA  33
     * (NRNGM1),WORK2(NRNGMA),WORK2(NYIELD),WORK2(NBURST),WORK2(NRVS),  QUANTA  34
     * WORK2(NMPROF),WORK2(NFMTIM),WORK2(NFMRNG),NWPNS,MAXAPB,MAXNMPS,  QUANTA  35
     * WORK1(NBASVA))                                                     QUANTA  36
C             CALCULATE THE REMAINING ADDRESSES IN WORK1 AND WORK2        QUANTA  37
      CALL REMW2(NWORK2,MAXAPB,MAXNMPS,NBRT,NTGTN,NLCM2,NRAN)             QUANTA  38
      CALL REMW1(NWORK1,NMARV,NSURV,NALOCD,NCOT,NLCM1,NWPNS,MAXAPB,NSRV) QUANTA  39
      NAF=3*NTGTS*NWPNS                                                   QUANTA  40
      WRITE(6,602) NAF,NLCM3                                              QUANTA  41
      IF(NAF.GT.NLCM3) STOP  "  ALOC, FLAMB DIMENSIONS "                  QUANTA  42
C            WORK2 ARRAY STORES VARIABLES USED IN THE COMPUTATION         QUANTA  43
C            OF THE SURVIVABILITIES;  CALCULATEE ADDRESSES TO BE USED     QUANTA  44
C            LATER FOR OTHER VARIABLES                                    QUANTA  45
      CALL SECW2 (NTGTS,NTYPES,NLCM2,NPROD,NVKILL,NSUMAK,NACT,NSUMV,      QUANTA  46
     * NSUMVK,NBONUS,NASHI,NASLO,NXLHI,NXLLO,MAXAPB,NPSB,MTYPES)         QUANTA  47
C             READ AIRCRAFT PROFILE DATA AND REMAINDER OF INPUT           QUANTA  48
      CALL PROCES(WORK2(NPSI),WORK2(NICAL),WORK2(NYIELD),NTYPES,MTYPES,  QUANTA  49
     * NPAP,NXAP,WORK2(NS),WORK2(ND),WORK2(NA),WORK2(NF),WORK2(NG),     QUANTA  50
     * WORK2(NVS),WORK2(NVEL),WORK2(NACCEL),WORK2(NNDATA),WORK2(NFLRP)) QUANTA  51
```

```
     * ,WORK2(NFTSA),WORK2(NFLRT),WORK2(NBURST))
      CALL MARVIN (NMARV,WORK1(NCUT),IMARV)
      CALL ALOCIN (NTGTS,NSUBS,NWPNS,MXNAPB,ALOC,WORK1(NALOCD),
     * WORK1(NISUBS),WORK1(NNMPS),WORK2(NITGTN))
      CALL LETHAL(WORK2(NGPTS),WORK2(NDTCEN),NTYPES,MTYPES,NPAP,NXAP,
     * WORK2(NS),WORK2(ND),WORK2(NA),WORK2(NF),WORK2(NNDATA),WORK2
     * (NFLRP),WORK2(NFTSA),WORK2(NFLRT),WORK2(NBURST),WORK2(NPSI),
     * WORK2(NICAL),WORK2(NYIELD),WORK2(NQAREA),WORK2(NQRLMX) )
C             ALL THE "KILL AREA" CALCULATIONS ARE COMPLETE
C             BEGIN COMPUTING SURVIVABILITIES
      CALL SRVMAT(WORK1(NSURV),WORK1(NSRV),WORK2(NBR(IM),WORK2(NBRI),
     * WORK2(NNRWAY),WORK1(NISEQ),WORK2(NDELTA),WORK2(NTGTLA),WORK2
     * (NTGTLN),WORK2(NSUBLA),WORK2(NSUBLN),WORK1(NMTYPE),WORK1(NREL),
     * WORK2(NRVS),WORK2(NRNUMI),WORK2(NRNUMA),WORK2(NMFRUF),WORK2
     * (NFMRNG),WORK2(NFMTIM),WORK1(NNMPS),WORK2(NDELTM),WORK2(NQAREA),
     * WORK2(NQRLMX),WORK2(NDTCEN),WORK2(NS),WORK2(ND),WORK2(NRAN),
     * WORK2(NGPTS),NWPNS,MXNAPB,NTYPES,MTYPES,NTGTS,NACT,MXRWAY,NSUBS,
     * MXMPR,NXAP,WORK1(NVAL))
C             COMPUTE THE OBJECTIVE FUNCTION
      CALL TGTKIL(NTGTS,NWPNS,NTYPES,ALOC,WORK1(NSURV),WORK1(NVAL),WORK1
     * (NRELVA),FLAMB,WORK2(NPROD),WORK2(NVKILL),MXNAPB,NACT,WORK1
     * (NISEQ),WORK1(NSRV),WORK2(NBONUS),WORK1(NBASVA),WORK1(NMTYPE),
     * WORK1(NREL),WORK1(NNMPS),NSUBS,MTYPES,WORK2(NPSB))
      CALL KILOUT(IMARV,WORK1(NCUT),NMARV,NTYPES,WORK2(NSUMV),WORK2
     * (NSUMVK),WORK2(NSUMAK),NTGTS,ALOC,WORK1(NVAL),WORK1(NISEQ),WORK2
     * (NPROD),NACT,WORK1(NRELVA),WORK2(NVKILL),OBJSUM,WORK1(NALOCD),
     * NWPNS,1,WORK1(NBASVA),WORK2(NBONUS))
C         CHECK FOR RESTART TAPE
      CALL RSTART(1,IMARV,OBJSUM,ALOC,FLAMB,WORK2(NPROD),WORK2(NVKILL),
     * WORK1(NALOCD),NTGTS,NWPNS,NACT)
C             FIND OPTIMAL ALLOCATION
   60 CALL REALOC(NTGTS,NWPNS,NSUBS,NTYPES,NMARV,IMARV,IOUT,WORK1(NSURV)
     * ,WORK1(NVAL),WORK1(NRELVA),WORK1(NISUBS),WORK1(NNMPS),ALOC,WORK1
     * (NALOCD),WORK1(NCUT),FLAMB,WORK2(NVKILL),WORK2(NPROD),OBJSUM,
     * WORK1(NISEQ),MXNAPB,NACT,WORK1(NBASVA),WORK2(NBONUS),WORK1
     * (NMTYPE),WORK1(NREL),MTYPES,WORK1(NSRV),WORK2(NASHI),WORK2
     * (NASLO),WORK2(NXLHI),WORK2(NXLLO),WORK2(NPSB))
C             OPTIMUM ATTAINED.  INTEGERIZE AND PRINT RESULTS
      CALL ALINT(NTGTS,NWPNS,ALOC,WORK1(NALOCD))
      CALL ALOUT(NTGTS,NSUBS,NWPNS,ALOC,WORK1(NISUBS),WORK1(NNMPS))
      CALL TGTKIL(NTGTS,NWPNS,NTYPES,ALOC,WORK1(NSURV),WORK1(NVAL),WORK1
     * (NRELVA),FLAMB,WORK2(NPROD),WORK2(NVKILL),MXNAPB,NACT,WORK1
     * (NISEQ),WORK1(NSRV),WORK2(NBONUS),WORK1(NBASVA),WORK1(NMTYPE),
     * WORK1(NREL),WORK1(NNMPS),NSUBS,MTYPES,WORK2(NPSB))
      CALL KILOUT(IMARV,WORK1(NCUT),NMARV,NTYPES,WORK2(NSUMV),WORK2
     * (NSUMVK),WORK2(NSUMAK),NTGTS,ALOC,WORK1(NVAL),WORK1(NISEQ),WORK2
     * (NPROD),NACT,WORK1(NRELVA),WORK2(NVKILL),OBJSUM,WORK1(NALOCD),
     * NWPNS,0,WORK1(NBASVA),WORK2(NBONUS))
C             CHECK FOR ANOTHER MARVAL
      IF (IQUIT.EQ.1. OR. ISSW1.EQ.1) CALL TERM (IQUIT,ISSW1)
```

```
        IF(IMARV.LE.NMARV .AND. NMARV.GT.0) GO TO 60                     QUANTA   102
C                                                                        QUANTA   103
        STOP                                                             QUANTA   104
  602   FORMAT(1X,* THE REQUIRED SIZE FOR THE ALOC AND FLAMB ARRAYS IS *, QUANTA  105
       * I7,/*  THE SIZE YOU HAVE DESIGNATED IS *,I7)                    QUANTA   106
  601   FORMAT (1X,* THE REQUIRED SIZE FOR THE INDX ARRAY IS *,I7/       QUANTA   107
       * *  THE SIZE YOU HAVE DESIGNATED IS *,I7)                        QUANTA   108
        END                                                              QUANTA   109
```

```
      SUBROUTINE WORKR1(NTGTS,NTYPES,NSUBS,NVAL,NRELVA,NISUBS,NNMPS,
     * NMTYPE,NWORK1,NLCM1,NISEG,NACT,NREL,MTYPES,NBASVA)
C               SUBROUTINE WORK1 CALCULATES SUBARRAY STARTING
C               ADDRESSES IN THE WORK1 ARRAY
      NVAL=1
      NRELVA=NVAL+NTGTS*NTYPES
      NISUBS=NRELVA+NTYPES
      NNMPS=NISUBS+NSUBS
      NMTYPE=NNMPS+NSUBS
      NREL=NMTYPE+NSUBS
      NISEG=NREL+MTYPES
      NBASVA=NISEG+NACT
      NWORK1=NBASVA+NTGTS
      IF((NWORK1-1).GT.NLCM1)WRITE(6,6000)NWORK1,NLCM1
      RETURN
 6000 FORMAT(/,36H SUBROUTINE WORKR1 HAS DETECTED THAT,/,
     * 43H THE REQUIRED WORK1 ARRAY SIZE IS AT LEAST ,I10,/,
     * 34H THE PROVIDED WORK1 ARRAY SIZE IS ,I10,/,
     * 60H HENCE AN ERROR WILL BE DETECTED LATER IF WE DONT BOMB FIRST)
      END
```

```
      SUBROUTINE REM#1(NWORK1,NMARV,NSORV,NALOCD,NCUT,NLCM1,NWPNS,MANAPB QUANTA   130
     * ,NSRV)                                                           QUANTA   131
C               SUBROUTINE REM#1 CALCULATES THE REMAINING SUBARRAY      QUANTA   132
C               STARTING ADDRESSES IN THE WORK1 ARRAY                   QUANTA   133
      DATA WHERE/5HREM#1/                                               QUANTA   134
      NSORV=NWORK1                                                      QUANTA   135
      NSRV=NSORV+NWPNS*MXNAPB                                           QUANTA   136
      NALOCD=NSRV+NWPNS*MXNAPB                                          QUANTA   137
      NCUT=NALOCD+NWPNS                                                 QUANTA   138
      NWORK1=NCUT+NMARV-1                                               QUANTA   139
      WRITE(6,6000)NWORK1,NLCM1                                         QUANTA   140
      IF(NWORK1.GT.NLCM1) CALL ABORT(WHERE,1,"SIZE OF WORK1 ARRAY EXCEED QUANTA   141
     IS DIMENSION")                                                     QUANTA   142
      RETURN                                                            QUANTA   143
 6000 FORMAT(34H THE REQUIRED WORK1 ARRAY SIZE IS ,I10,/,               QUANTA   144
     * 34H THE PROVIDED WORK1 ARRAY SIZE IS ,I10)                       QUANTA   145
      END                                                               QUANTA   146
```

PAGE 6

```
      SUBROUTINE SECW2(NTGTS,NTYPES,NLCM2,NPROD,NVKILL,NSUMAK,NACT,      QUANTA   147
     * NSUMV,NSUMVK,NBONUS,NASHI,NASLO,NALHI,NALLO,MANAPB,NPSB,NTYPES)   QUANTA   148
C               SUBROUTINE SECW2 CALCULATES SUBARRAY STARTING            QUANTA   149
C               ADDRESSES IN THE WORK2 ARRAY                             QUANTA   150
      DATA WHERE/5HSECW2/                                                QUANTA   151
C                                                                        QUANTA   152
C               CALCULATE STARTING ADDRESSES                             QUANTA   153
      NPROD=1                                                            QUANTA   154
      NVKILL=NPROD+NACT                                                  QUANTA   155
      NSUMAK=NVKILL+NTGTS                                                QUANTA   156
      NSUMV=NSUMAK+NTYPES                                                QUANTA   157
      NSUMVK=NSUMV+NTYPES                                                QUANTA   158
      NBONUS=NSUMVK+NTYPES                                               QUANTA   159
      NASHI = NBONUS + NTGTS                                             QUANTA   160
      NASLO = NASHI + MANAPB                                             QUANTA   161
      NALHI = NASLO + MANAPB                                             QUANTA   162
      NALLO = NALHI + MANAPB                                             QUANTA   163
      NPSB=NALLO+MANAPB                                                  QUANTA   164
      NWORK2=NPSB+NTYPES-1                                               QUANTA   165
      WRITE(6,6000)NWORK2,NLCM2                                          QUANTA   166
      IF(NWORK2.GT.NLCM2) CALL ABORT(WHERE,1,"SIZE OF WORK2 ARRAY EXCEED QUANTA   167
     1S DIMENSION")                                                      QUANTA   168
      RETURN                                                             QUANTA   169
 6000 FORMAT(34H THE REQUIRED WORK2 ARRAY SIZE IS ,I10,/,                QUANTA   170
     * 34H THE PROVIDED WORK2 ARRAY SIZE IS ,I10)                        QUANTA   171
      END                                                                QUANTA   172
```

```
      SUBROUTINE WORKR2(NTOTS,NSUBS,NTYPES,MARWAY,MTYPES,NPAP,NAAP,      QUANTA   173
     * MAMPR,NLCM2,NTOTLA,NTOTLN,NDTCEN,NNRWAY,NBRTIM,NPSI,NICAL,NDELTA, QUANTA   174
     * NSUBLA,NSUBLN,NDELTM,NRNGMI,NRNGMA,NYIELD,NBURST,NRVS,NMPROF,     QUANTA   175
     * NFMTIM,NFMRNG,NOPTS,NGAREA,NGRLMA,NS,ND,NA,NF,NG,NVS,NVEL,NACCEL, QUANTA   176
     * NNDATA,NFLRP,NFTSA,NFLRT,NWORK2)                                  QUANTA   177
C               SUBROUTINE WORKR2 CALCULATES SUBARRAY STARTING           QUANTA   178
C               ADDRESSES IN THE WORK2 ARRAY                             QUANTA   179
      NTOTLA=1                                                           QUANTA   180
      NTOTLN=NTOTLA+NTOTS                                                QUANTA   181
      NDTCEN=NTOTLN+NTOTS                                                QUANTA   182
      NNRWAY=NDTCEN+NTYPES                                               QUANTA   183
      NBRTIM=NNRWAY+NTOTS                                                QUANTA   184
      NPSI=NBRTIM+NTYPES                                                 QUANTA   185
      NICAL=NPSI+NTYPES                                                  QUANTA   186
      NDELTA=NICAL+NTYPES                                                QUANTA   187
      NSUBLA=NDELTA+MARWAY*NTYPES*NTYPES                                 QUANTA   188
      NSUBLN=NSUBLA+NSUBS                                                QUANTA   189
      NDELTM=NSUBLN+NSUBS                                                QUANTA   190
      NRNGMI=NDELTM+MTYPES                                               QUANTA   191
      NRNGMA=NRNGMI+MTYPES                                               QUANTA   192
      NYIELD=NRNGMA+MTYPES                                               QUANTA   193
      NBURST=NYIELD+MTYPES                                               QUANTA   194
      NRVS=NBURST+MTYPES                                                 QUANTA   195
      NMPROF=NRVS+MTYPES                                                 QUANTA   196
      NFMTIM=NMPROF+MTYPES                                               QUANTA   197
      NFMRNG=NFMTIM+MAMPR*MTYPES                                         QUANTA   198
      NOPTS=NFMRNG+MAMPR*MTYPES                                          QUANTA   199
      NGAREA=NOPTS+30                                                    QUANTA   200
      NGRLMA=NGAREA +30*NTYPES*MTYPES                                    QUANTA   201
      NS=NGRLMA+30*NTYPES*MTYPES                                         QUANTA   202
      ND=NS+NAAP*NTYPES                                                  QUANTA   203
      NA=ND+NAAP*NTYPES                                                  QUANTA   204
      NF=NA+NPAP*NTYPES                                                  QUANTA   205
      NG=NF+NPAP*NTYPES                                                  QUANTA   206
      NVS=NG+NPAP*NTYPES                                                 QUANTA   207
      NVEL=NVS+NPAP*NTYPES                                               QUANTA   208
      NACCEL=NVEL+NPAP*NTYPES                                            QUANTA   209
      NNDATA=NACCEL+NPAP*NTYPES                                          QUANTA   210
      NFLRP=NNDATA+NTYPES                                                QUANTA   211
      NFTSA=NFLRP+NTYPES*MTYPES                                          QUANTA   212
      NFLRT=NFTSA+NTYPES*MTYPES                                          QUANTA   213
      NWORK2=NFLRT+NTYPES*MTYPES                                         QUANTA   214
      IF(NWORK2-1.GT.NLCM2) WRITE (6,10) NWORK2                          QUANTA   215
      RETURN                                                             QUANTA   216
   10 FORMAT(/* SUBROUTINE WORKR2 HAS DETECTED THAT THE REQUIRED WORK2   QUANTA   217
     * ARRAY SIZE IS*,I10/* INCREASE THE DIMENSION OF THE WORK2 ARRAY TO QUANTA   218
     *  AT LEAST THAT AMOUNT.*)                                          QUANTA   219
      END                                                                QUANTA   220
```

```
      SUBROUTINE REMW2(NWORK2,MANAPB,MANMPS,NBRT,NITGTN,NLCM2,NRAN)      QUANTA  221
C             SUBROUTINE REMW2 CALCULATES THE STARTING ADDRESSES OF     QUANTA  222
C             THE REMAINING SUBARRAYS IN THE WORK2 ARRAY                QUANTA  223
      DATA WHERE/5HREMW2/                                               QUANTA  224
      NBRT=NWORK2                                                       QUANTA  225
      NITGTN=NBRT+MANAPB                                                QUANTA  226
      NRAN=NITGTN+MANMPS                                                QUANTA  227
      NWORK2=NRAN+MANAPB-1                                              QUANTA  228
      WRITE(6,6000)NWORK2,NLCM2                                         QUANTA  229
      IF(NWORK2.GT.NLCM2) CALL ABORT(WHERE,1,"SIZE OF WORK2 ARRAY EXCEED QUANTA  230
     IS DIMENSION")                                                     QUANTA  231
      RETURN                                                            QUANTA  232
 6000 FORMAT(/,34H THE REQUIRED WORK2 ARRAY SIZE IS ,I10,/,             QUANTA  233
     * 34H THE PROVIDED WORK2 ARRAY SIZE IS ,I10)                       QUANTA  234
      END                                                               QUANTA  235
```

```
      SUBROUTINE ABORT (WHERE,NUM,MSG)                                  QUANTA 236
      DIMENSION MSG(12)                                                 QUANTA 237
C                                                                       QUANTA 238
      WRITE (6,3) NUM,WHERE                                             QUANTA 239
      DO 1 L=1,12                                                       QUANTA 240
      IF (MSG(L).EQ.0) GO TO 2                                          QUANTA 241
    1 CONTINUE                                                          QUANTA 242
      L=13                                                              QUANTA 243
C                                                                       QUANTA 244
    2 L=L-1                                                             QUANTA 245
      IF (L.GT.0) WRITE (6,4) (MSG(I),I=1,L)                            QUANTA 246
C                                                                       QUANTA 247
      STOP "PROGRAM ABORT "                                             QUANTA 248
C                                                                       QUANTA 249
    3 FORMAT (1X,72H *****   ABORT NUMBER,I3,13H CALLED FROM ,A6,3X,5H* QUANTA 250
     1****)                                                             QUANTA 251
    4 FORMAT (1X,5H*****,3X,12A10)                                      QUANTA 252
      END                                                               QUANTA 253
```

```
      SUBROUTINE STARTR(NTGTS,NSUBS,NTYPES,MARWAY,MTYPES,IOUT,NACT,NMARV QUANTA 254
     1 )                                                                 QUANTA 255
      DATA WHERE/6HSTARTR/                                               QUANTA 256
C               READ HEADER CARD                                         QUANTA 257
      READ(5,5000) NTGTS,NSUBS,NTYPES,MARWAY,MTYPES,IOUT,NACT,NMARV      QUANTA 258
      CALL PAGE(0)                                                       QUANTA 259
      WRITE(6,6010) NMARV,NTGTS,NSUBS,NTYPES,NACT,MTYPES                 QUANTA 260
      IF(NTGTS.LE.0) CALL ABORT(WHERE,1,"NUMBER OF BASES MUST BE GREATER QUANTA 261
     1 THAN 0")                                                          QUANTA 262
      IF(NSUBS.LE.0) CALL ABORT(WHERE,2,"NUMBER OF SUBS MUST BE GREATER  QUANTA 263
     1THAN 0")                                                           QUANTA 264
      IF(NTYPES.LE.0) CALL ABORT(WHERE,3,"NUMBER OF AIRCRAFT TYPES MUST  QUANTA 265
     1BE GREATER THAN 0")                                                QUANTA 266
      IF(MTYPES.LE.0) CALL ABORT(WHERE,4,"NUMBER OF MISSILE TYPES MUST B QUANTA 267
     1E GREATER THAN 0")                                                 QUANTA 268
      IF(MARWAY.LE.0) CALL ABORT(WHERE,5,"MAXIMUM NUMBER OF RUNWAYS ON A QUANTA 269
     1 BASE MUST BE GREATER THAN 0")                                     QUANTA 270
      IF((IOUT.NE.1).AND.(IOUT.NE.2)) CALL ABORT(WHERE,6,"IOUT MUST BE E QUANTA 271
     1ITHER 1 OR 2")                                                     QUANTA 272
      IF(NACT.LE.0) CALL ABORT(WHERE,7,"THE TOTAL NUMBER OF AIRCRAFT MUS QUANTA 273
     *T BE GREATER THAN 0")                                              QUANTA 274
      IF(NMARV.LT.0) CALL ABORT(WHERE,8,"THE NUMBER OF MARVAL RUNS CANNO QUANTA 275
     *T BE NEGATIVE")                                                    QUANTA 276
      CALL PAGE(2)                                                       QUANTA 277
      RETURN                                                             QUANTA 278
 5000 FORMAT(8I5)                                                        QUANTA 279
 6010 FORMAT(1X,*THERE ARE *,I5,* MARVAL RUNS FOR THIS PROBLEM WITH *,   QUANTA 280
     * I5,* TARGETS, *,I5,* SUB LOCATIONS, *,/10X,I5,* TYPES OF AIRCRAFT QUANTA 281
     * (TOTAL OF *,I5,* ), AND *,I5,* TYPES OF MISSILES.*)               QUANTA 282
      END                                                                QUANTA 283
```

```
      SUBROUTINE READER(NTGTS,NSUBS,NTYPES,MTYPES,MAXWAY,NACT,MAMPR,VAL,
     * RELVAL,ISUBS,NMPS,MTYPE,ISEQ,TGTLAT,TGTLNG,DTCENT,NRWAYS,BRTIME,
     * PSI,CAL,DELTAC,SUBLAT,SUBLNG,DELTM,REL,RNGMIN,RNGMAX,YIELD,BURST,
     *RVS,MPROF,FMTIME,FMRNG,NWPNS,MXNAPB,MXNMPS,BASVAL)
      DIMENSION VAL(NTGTS,NTYPES),RELVAL(NTYPES),ISUBS(NSUBS),NMPS(NSUBS
     *),MTYPE(NSUBS),ISEQ(NACT),TGTLAT(NTGTS),TGTLNG(NTGTS),
     * NRWAYS(NTGTS),BRTIME(NTYPES),PSI(NTYPES),CAL(NTYPES),DELTAC(MAX
     *WAY,NTYPES,NTYPES),SUBLAT(NSUBS),SUBLNG(NSUBS),DELTM(MTYPES),REL(M
     *TYPES),RNGMIN(MTYPES),RNGMAX(MTYPES),YIELD(MTYPES),BURST(MTYPES),R
     *VS(MTYPES),MPROF(MTYPES),FMTIME(MAMPR,MTYPES),FMRNG(MAMPR,MTYPES)
     *,DTCENT(NTYPES),BASVAL(NTGTS)
      LEVEL 2,VAL,RELVAL,ISUBS,NMPS,MTYPE,ISEQ,TGTLAT,TGTLNG,DTCENT,
     * NRWAYS,BRTIME,PSI,CAL,DELTAC,SUBLAT,SUBLNG,DELTM,REL,RNGMIN,RNGMA
     *X,YIELD,BURST,RVS,MPROF,FMTIME,FMRNG,BASVAL
      DATA RDPDEG/.01745329251/
      DATA WHERE/6HREADER/
C             READ AIRCRAFT BEDDOWN CARDS
      NAC=0
      MANAPB=0
      DO 20 I=1,NTGTS
      READ(5,5000)(TGTLAT(I),TGTLNG(I),BASVAL(I),NRWAYS(I),
     * (VAL(I,J),J=1,NTYPES))
      NAC1=NAC+1
      DO 10 J=1,NTYPES
      NAC=NAC+INT(VAL(I,J)+.01)
   10 CONTINUE
      IF(NAC.GT.NACT) WRITE(6,31) NAC,NACT
   31 FORMAT(1X,*QUANTA FOUND THE TOTAL NUMBER OF AIRCRAFT TO BE*,I6,*
     1  THE ORIGINAL INPUT VALUE IS *,I6)
      IF(NAC.GT.NACT) CALL ABORT(WHERE,1,"TOTAL NUMBER OF AIRCRAFT IS GR
     1EATER THAN ORIGINAL INPUT")
      READ(5,5010)(ISEQ(IAC),IAC=NAC1,NAC)
      NAPB=NAC-NAC1+1
      MANAPB=MAX0(MANAPB,NAPB)
   20 CONTINUE
C              READ DISTANCE TO CENTROID FOR EACH AIRCRAFT TYPE
      READ(5,5030) (DTCENT(J),J=1,NTYPES)
C              READ AIRCRAFT TYPE CARDS
      READ(5,5020)(RELVAL(J),BRTIME(J),PSI(J),CAL(J),J=1,NTYPES)
C              READ TAKEOFF INTERVAL MATRIX
      READ(5,5030)(((DELTAC(IRW,JLD,KFL),KFL=1,NTYPES),JLD=1,NTYPES),
     * IRW=1,MAXWAY)
C              WRITE TARGET INFO
      WRITE(6,6000)
      CALL PAGE(3)
      NAC=0
      DO 40 I=1,NTGTS
      NAC1=NAC+1
      DO 30 J=1,NTYPES
      NAC=NAC+INT(VAL(I,J)+.01)
```

```
   30 CONTINUE                                                          QUANTA   334
      FNAC=FLOAT(NAC-NAC1)+1.0                                          QUANTA   335
      NLINES=(FNAC-1.)/30.                                              QUANTA   336
      CALL PAGE(-MAX0(2,NTYPES)-NLINES-2)                               QUANTA   337
      WRITE(6,6010)I,BASVAL(I),TOTLAT(I),TOTLNG(I),NRWAYS(I),(J,VAL(I,J)QUANTA   338
     *,J=1,NTYPES)                                                      QUANTA   339
      IF((TOTLAT(I).LT.0.0).OR.(TOTLNG(I).LT.0.0)) CALL ABORT(WHERE,2," QUANTA   340
     1BASE LATITUDE AND LONGITUDE MUST BE GREATER THAN 0")              QUANTA   341
      IF(NRWAYS(I).LE.0 .OR. NRWAYS(I).GT.MXRWAY) CALL ABORT(WHERE,3,"NUQUANTA   342
     *MBER OF RUNWAYS ON A BASE MUST BE GREATER THAN 0 AND NOT GREATER TQUANTA   343
     1HAN THE MAXIMUM NUMBER OF RUNWAYS")                               QUANTA   344
      DO 35 J=1,NTYPES                                                  QUANTA   345
      IF(VAL(I,J).LT.0.0) CALL ABORT(WHERE,4,"NUMBER OF AIRCRAFT OF A GIQUANTA   346
     1VEN TYPE ON A GIVEN BASE CANNOT BE LESS THAN 0")                  QUANTA   347
   35 CONTINUE                                                          QUANTA   348
      WRITE(6,6020)(ISEQ(IAC),IAC=NAC1,NAC)                             QUANTA   349
      DO 37 IAC=NAC1,NAC                                                QUANTA   350
      IF((ISEQ(IAC).LE.0).OR.(ISEQ(IAC).GT.NTYPES)) CALL ABORT(WHERE,5," QUANTA   351
     1AN AIRCRAFT TYPE ON A BASE MUST BE GREATER THAN 0 AND NOT GREATER QUANTA   352
     2THAN THE NUMBER OF TYPES")                                        QUANTA   353
   37 CONTINUE                                                          QUANTA   354
   40 CONTINUE                                                          QUANTA   355
C              WRITE AIRCRAFT TYPE PARAMETERS                           QUANTA   356
      CALL PAGE(-NTYPES-3)                                              QUANTA   357
      WRITE(6,6030)(J,RELVAL(J),BRTIME(J),DTCENT(J),PSI(J),CAL(J),J=1,N QUANTA   358
     *TYPES)                                                            QUANTA   359
      DO 45 J=1,NTYPES                                                  QUANTA   360
      IF(PSI(J).LE.0.0) CALL ABORT(WHERE,6,"AIRCRAFT OVERPRESSURE MUST  QUANTA   361
     *BE GREATER THAN 0")                                               QUANTA   362
      IF(CAL(J).LE.0.0) CALL ABORT(WHERE,7,"THE VULNERABILITY LEVEL OF TQUANTA   363
     1HE AIRCRAFT TO FREE-FIELD THERMAL ENERGY CANNOT BE LESS THAN OR EQQUANTA   364
     2UAL TO 0")                                                        QUANTA   365
      IF(DTCENT(J).LE.0.0) CALL ABORT(WHERE,8,"THE DISTANCE FROM BRAKE RQUANTA   366
     1ELEASE POINT TO THE TURN POINT MUST BE GREATER THAN 0")           QUANTA   367
      IF(BRTIME(J).LE.0.0) CALL ABORT(WHERE,9,"THE BRAKE RELEASE TIME MUQUANTA   368
     1ST BE GREATER THAN 0")                                            QUANTA   369
   45 CONTINUE                                                          QUANTA   370
C              WRITE TAKEOFF INTERVALS MATRIX                           QUANTA   371
      CALL PAGE(-MARWAY*NTYPES*NTYPES-3)                                QUANTA   372
      WRITE(6,6040)(((IRW,JLD,KFL,DELTAC(IRW,JLD,KFL),KFL=1,NTYPES),    QUANTA   373
     * JLD=1,NTYPES),IRW=1,MXRWAY)                                      QUANTA   374
      DO 47 IRW=1,MXRWAY                                                QUANTA   375
      DO 47 JLD=1,NTYPES                                                QUANTA   376
      DO 47 KFL=1,NTYPES                                                QUANTA   377
      IF(DELTAC(IRW,JLD,KFL).LE.0.0) CALL ABORT(WHERE,10,"THE TAKE-OFF IQUANTA   378
     1NTERVALS BETWEEN AIRCRAFT TYPES MUST BE GREATER THAN 0")          QUANTA   379
   47 CONTINUE                                                          QUANTA   380
C              READ SUB LOCATION CARDS                                  QUANTA   381
      NWPNS=0                                                           QUANTA   382
      MXNMPS=0                                                          QUANTA   383
```

```
      DO 50 I=1,NSUBS                                                   QUANTA   384
      READ(5,5040)SUBLAT(I),SUBLNG(I),ISUBS(I),NMPS(I),MTYPE(I)         QUANTA   385
      MXNMPS=MAX0(MXNMPS,NMPS(I))                                       QUANTA   386
      NWPNS=NWPNS+NMPS(I)                                               QUANTA   387
   50 CONTINUE                                                          QUANTA   388
C              WRITE SUB LOCATION CARDS                                 QUANTA   389
      CALL PAGE(-NSUBS-2)                                               QUANTA   390
      WRITE(6,6050)(I,SUBLAT(I),SUBLNG(I),ISUBS(I),NMPS(I),MTYPE(I),    QUANTA   391
     *I=1,NSUBS)                                                        QUANTA   392
      DO 55 I=1,NSUBS                                                   QUANTA   393
      IF((SUBLAT(I).LT.0.0).OR.(SUBLNG(I).LT.0.0))CALL ABORT(WHERE,11,"T QUANTA   394
     1HE SUB LATITUDE AND LONGITUDE MUST BE GREATER THAN 0")            QUANTA   395
      IF(ISUBS(I).LE.0) CALL ABORT(WHERE,12,"THE NUMBER OF SUBS AT A GIV QUANTA   396
     1EN LOCATION MUST BE GREATER THAN 0")                              QUANTA   397
      IF(NMPS(I).LE.0) CALL ABORT(WHERE,13,"THE NUMBER OF MISSILES ON AL QUANTA   398
     1L SUBS AT A GIVEN LOCATION MUST BE GREATER THAN 0")               QUANTA   399
      IF((MTYPE(I).LE.0).OR.(MTYPE(I).GT.MTYPES))CALL ABORT(WHERE,14,"TH QUANTA   400
     1E TYPE OF SUB MUST BE GREATER THAN 0 AND NOT GREATER THAN THE TOTA QUANTA   401
     2L NUMBER OF SUB TYPES")                                           QUANTA   402
   55 CONTINUE                                                          QUANTA   403
C              CALCULATE AND WRITE TOTAL NUMBER OF SUBS OF EACH TYPE    QUANTA   404
      CALL PAGE(-MTYPES*2)                                              QUANTA   405
      DO 70 J=1,MTYPES                                                  QUANTA   406
      NSUBT=0                                                           QUANTA   407
      DO 60 I=1,NSUBS                                                   QUANTA   408
      IF(MTYPE(I).EQ.J)NSUBT=NSUBT+ISUBS(I)                             QUANTA   409
   60 CONTINUE                                                          QUANTA   410
      WRITE(6,6060)J,NSUBT                                              QUANTA   411
   70 CONTINUE                                                          QUANTA   412
C              READ SUB TYPE PARAMETERS AND PROFILES                    QUANTA   413
      DO 80 I=1,MTYPES                                                  QUANTA   414
      READ(5,5030)DELTM(I),REL(I),RNGMIN(I),RNGMAX(I),YIELD(I),BURST(I), QUANTA   415
     *RVS(I)                                                            QUANTA   416
      READ(5,5060)MPR,(FMTIME(J,I),FMRNG(J,I),J=1,MPR)                  QUANTA   417
      IF(MPR.GT.MXMPR) CALL ABORT(WHERE,15,"NUMBER OF TIME/DISTANCE PAIR QUANTA   418
     1S CANNOT BE GREATER THAN MXMPR")                                  QUANTA   419
      MPROF(I)=MPR                                                      QUANTA   420
   80 CONTINUE                                                          QUANTA   421
C              WRITE SUB TYPE PARAMETERS AND PROFILES                   QUANTA   422
      CALL PAGE(-MTYPES-3)                                              QUANTA   423
      WRITE(6,6070)(J,DELTM(J),REL(J),RNGMIN(J),RNGMAX(J),YIELD(J),BURST QUANTA   424
     *(J),RVS(J),J=1,MTYPES)                                            QUANTA   425
      DO 85 J=1,MTYPES                                                  QUANTA   426
      IF(DELTM(J).LE.0.0) CALL ABORT(WHERE,16,"THE TIME INTERVAL BETWEEN QUANTA   427
     * SALVO LAUNCHES MUST BE GREATER THAN 0")                          QUANTA   428
      IF((REL(J).LE.0.0).OR.(REL(J).GE.1.0))CALL ABORT(WHERE,17,"THE REL QUANTA   429
     1IABILITY OF A MISSILE MUST BE GREATER THAN 0 AND LESS THAN 1")    QUANTA   430
      IF(RNGMAX(J).LE.RNGMIN(J)) CALL ABORT(WHERE,18,"THE MAXIMUM RANGE QUANTA   431
     1OF THE MISSILE MUST BE GREATER THAN ITS MINIMUM RANGE")           QUANTA   432
      IF(YIELD(J).LE.0.0) CALL ABORT(WHERE,19,"THE YIELD OF THE MISSILE QUANTA   433
```

```
     1WARHEAD MUST BE GREATER THAN 0")
      IF(BURST(J).LT.0.0) CALL ABORT(WHERE,20,"THE BURST HEIGHT OF THE M
     1ISSILE CANNOT BE LESS THAN 0")
      IF(RVS(J).LT.1.0) CALL ABORT(WHERE,21,"THE NUMBER OF REENTRY VEHIC
     1LES CANNOT BE LESS THAN 1")
   85 CONTINUE
      DO 90 J=1,MTYPES
      MPR=MPROF(J)
      CALL PAGE(-MPR-2)
      WRITE(6,6080)J,(FMTIME(I,J),FMRNG(I,J),I=1,MPR)
      DO 87 I=1,MPR
      IF(FMTIME(I,J).LT.0.0) CALL ABORT(WHERE,22,"THE FLIGHT TIME OF THE
     1 MISSILE MUST BE GREATER THAN 0")
      IF(FMRNG(I,J).LT.0.0) CALL ABORT(WHERE,23,"THE DISTANCE CORRESPOND
     1ING TO THE MISSILE FLIGHT TIME MUST BE GREATER THAN 0")
   87 CONTINUE
      IF(FMRNG(MPR,J).NE.RNGMAX(J)) CALL ABORT(WHERE,24,"THE LAST RANGE
     *MUST BE EQUAL TO THE MAX")
   90 CONTINUE
C              CONVERT TARGET AND SUB LOCATIONS FROM DEGREES TO RADIANS
      DO 100 I=1,NTGTS
      TGTLAT(I)=TGTLAT(I)*RDPDEG
      TGTLNG(I)=TGTLNG(I)*RDPDEG
  100 CONTINUE
      DO 110 I=1,NSUBS
      SUBLAT(I)=SUBLAT(I)*RDPDEG
      SUBLNG(I)=SUBLNG(I)*RDPDEG
  110 CONTINUE
      RETURN
 5000 FORMAT(3F10.4,I5,5X,3F10.4/(7F10.4))
 5010 FORMAT(7011)
 5020 FORMAT (4F10.4)
 5030 FORMAT(7F10.4)
 5040 FORMAT(2F10.4,3I5)
 5060 FORMAT(I5,5X,6F10.4/(6F10.4))
 6000 FORMAT (/7H TARGET,4X,26H VALUE  LOCATION (DEGREES),3X,7HRUNWAYS
     1,2X,8HAIRCRAFT/53X,15HTYPE AND NUMBER)
 6010 FORMAT(/I7,3F10.4,2I10,F11.4/(47X,I10,F11.4))
 6020 FORMAT (32X,25HTAKE-OFF SEQUENCE BY TYPE,5X,30I2/(62X,30I2))
 6030 FORMAT (/2X,8HAIRCRAFT,2X,8HRELATIVE,5X,5HBRAKE,2X,8HCENTROID,/6X,
     *4HTYPE,5X,5HVALUE,3X,7HRELEASE,2X,8HDISTANCE,7X,3HPSI,7X,3HCAL,/(I
     *10,5F10.4))
 6040 FORMAT (/28H AIRCRAFT TAKE-OFF INTERVALS/8X,7HRUNWAYS,5X,5HTYPE1,5
     1X,5HTYPE2,3X,7HMINUTES/(I15,2I10,F10.4))
 6050 FORMAT (/2X,23HSUB LOCATIONS (DEGREES),6X,4HSUBS,3X,13HMISSILES AN
     1D ,4HTYPE/(I5,2F10.4,I10,I11,I8))
 6060 FORMAT (/23H NUMBER OF SUBS OF TYPE,I5,2H =,I5)
 6070 FORMAT(/3X,7HMISSILE,4X,6HLAUNCH,25X,3HMIN,7X,3HMAX,17X,6HHEIGHT,
     *8X,6HNUMBER,/,6X,4HTYPE,2X,8HINTERVAL,4X,11HRELIABILITY,10X,
     *5HRANGE,5X,5HRANGE,5X,5HYIELD,5X,8HOF BURST,4X,11HRVS/MISSILE,/,
```

```
     +I10,F10.4,3X,F8.4,9X,F10.3,F10.3,F10.4,4X,F10.2,10X,F4.1)         QUANTA     484
 6080 FORMAT (/13H MISSILE TYPE,11X,4HTIME,5X,5HRANGE/I13,5X,2F10.4/(18X QUANTA     485
     1,2F10.4))                                                         QUANTA     486
      END                                                               QUANTA     487
```

```
      SUBROUTINE PROCES(PSI,CAL,YIELD,NTYPES,MTYPES,NPAP,NXAP,S,D,      QUANTA
     * A,F,G,VS,VEL,ACCEL,NDATA,FLRP,FTSA,FLRT,BURST)                   QUANTA
      DIMENSION PSI(NTYPES),CAL(NTYPES),YIELD(MTYPES),S(NXAP,NTYPES),   QUANTA
     * D(NXAP,NTYPES),A(NPAP,NTYPES),F(NPAP,NTYPES),G(NPAP,NTYPES),VS   QUANTA
     * (NPAP,NTYPES),VEL(NPAP,NTYPES),ACCEL(NPAP,NTYPES),NDATA(NTYPES), QUANTA
     * FLRP(NTYPES,MTYPES),FTSA(NTYPES,MTYPES),FLRT(NTYPES,MTYPES),     QUANTA
     * BURST(MTYPES)                                                    QUANTA
C                                                                       QUANTA
C     THIS SUBROUTINE IS USED TO INPUT DATA FOR AIRCRAFT AND IS USED    QUANTA
C     TO GENERATE NUCLEAR EFFECTS GEOMETRY FOR SPECIFIED HARDNESS       QUANTA
C     LEVELS VIA SUBROUTINES SPARK AND SNARE                            QUANTA
C                                                                       QUANTA
      DIMENSION DUM(5)                                                  QUANTA
      COMMON/PASS/RHO,VIS,PZ,HTE                                        QUANTA
      LOGICAL SNNEW,VALID                                               QUANTA
      LEVEL 2, PSI,CAL,YIELD,D,S,A,F,G,VS,VEL,ACCEL,NDATA,              QUANTA
     * FLRP,FTSA,FLRT,BURST                                             QUANTA
      DATA DELT,CMTF,CFTM /1.0001,3.2808,0.3048/                        QUANTA
      DATA WHERE/6HPROCES/                                              QUANTA
      DO 130 I=1,NTYPES                                                 QUANTA
      CALL PAGE (0)                                                     QUANTA
      READ (5,500)AZ                                                    QUANTA
      READ (5,510)N                                                     QUANTA
      NDATA(I)=N                                                        QUANTA
      IF(AZ.LE.0.0) CALL ABORT(WHERE,1,"THE LEVEL-OFF ALTITUDE OF THE AI QUANTA
     1RCRAFT MUST BE GREATER THAN 0")                                   QUANTA
      IF(N.GT.NPAP) CALL ABORT(WHERE,2,"THE NUMBER OF DATA POINTS IN AIR QUANTA
     1CRAFT PROFILE IS LARGER THAN NPAP")                               QUANTA
      ICAL=CAL(I)                                                       QUANTA
      WRITE (6,600)I,NTYPES                                             QUANTA
      WRITE (6,610) PSI(I),ICAL                                         QUANTA
      WRITE (6,620)                                                     QUANTA
      CALL PAGE (17)                                                    QUANTA
C                                                                       QUANTA
C     THE FOLLOWING READS AIRCRAFT PROFILE INPUT DATA AND ADJUSTS THE   QUANTA
C     ALTITUDE(IF NECESSARY) FOR USE BY AOSC WHICH REQUIRES IT TO BE    QUANTA
C     STRICTLY MONOTONICALLY INCREASING.  A-ALTITUDE (FEET);            QUANTA
C     F-GROUND RANGE (FEET);  G-TIME (SEC);  VEL-MACH NUMBER;           QUANTA
C     ACCEL-ACCELERATION (FEET/SEC/SEC).  THE VELOCITY OF SOUND (VS)    QUANTA
C     GIVEN ALTITUDE IS CALCULATED BY FUNCTION SSPF.                    QUANTA
C                                                                       QUANTA
      READ (5,520)  F(1,I),G(1,I),A(1,I),VEL(1,I),ACCEL(1,I)            QUANTA
      IF(A(1,I).LT.0.0) CALL ABORT(WHERE,3,"THE AIRCRAFT ALTITUDE CANNOT QUANTA
     1 BE LESS THAN 0")                                                 QUANTA
      IF(F(1,I).LT.0.0) CALL ABORT(WHERE,4,"THE AIRCRAFT GROUND RANGE FR QUANTA
     1OM THE BRAKE RELEASE POINT CANNOT BE LESS THAN 0")                QUANTA
      IF(G(1,I).LT.0.0) CALL ABORT(WHERE,5,"THE TIME FROM THE BRAKE RELE QUANTA
     1ASE POINT CANNOT BE LESS THAN 0")                                 QUANTA
      ALT=CFTM*A(1,I)                                                   QUANTA
      VS(1,I)=CMTF*SSPF(ALT)                                            QUANTA
```

```
      J=1
      WRITE (6,630) J,F(1,I),G(1,I),A(1,I),VS(1,I),VEL(1,I),ACCEL(1,I)
      DO 30 J=2,N
      READ (5,520) F(J,I),G(J,I),A(J,I),VEL(J,I),ACCEL(J,I)
   10 IF (A(J,I).GT.A(J-1,I)) GO TO 20
      A(J,I)=DELT*A(J-1,I)
      IF (A(J,I) .LE. 0.0) A(J,I)=A(J-1,I)+0.001
      IF (AZ.EQ.A(J-1,I)) AZ=A(J,I)
      GO TO 10
   20 ALT=CFTM*A(J,I)
      VS(J,I)=CMTF*SSPF(ALT)
      WRITE (6,630) J,F(J,I),G(J,I),A(J,I),VS(J,I),VEL(J,I),ACCEL(J,I)
      IF(F(J,I).LE.F(J-1,I)) CALL ABORT(WHERE,6,"THE GROUND RANGE MUST B
     1E STRICTLY INCREASING")
      IF(G(J,I).LE.G(J-1,I)) CALL ABORT(WHERE,7,"TIME MUST BE STRICTLY I
     1NCREASING")
      CALL PAGE (1)
   30 CONTINUE
C
      READ (5,520)ALTFM,FMACH,TINT,XATI
      IF(ALTFM.LE.0.0) CALL ABORT(WHERE,8,"ALTITUDE AT FINAL MACH MUST B
     1E GREATER THAN 0")
      IF(FMACH.LE.0.0) CALL ABORT(WHERE,9,"FINAL MACH MUST BE POSITIVE")
      IF(TINT.LE.0.0) CALL ABORT(WHERE,10,"TIME INTERVAL MUST BE POSITIV
     1E")
      IF(XATI.LT.0.0) CALL ABORT(WHERE,11,"THE NUMBER OF DATA POINTS MUS
     1T BE POSITIVE")
C
      READ (5,500) HTE,RHO,VIS,PZ
      IF(RHO.LT.0.0.OR.RHO.GT.1.0) CALL ABORT(WHERE,12,"GROUND REFLECTAN
     1CE MUST BE BETWEEN 0 AND 1")
      IF(VIS.LT.0.0) CALL ABORT(WHERE,13,"VISIBILITY MUST BE POSITIVE")
      IF(PZ.LT.0.0) CALL ABORT(WHERE,14,"WATER VAPOR MUST BE POSITIVE")
      IF(HTE.LT.0.0)  CALL ABORT(WHERE,15,"THE TERRAIN HEIGHT OF THE AIR
     1CRAFT CANNOT BE LESS THAN 0")
C
C     S AND D ARE TIME/DISTANCE POINTS IN MINUTES/NM IN AN EXTENDED
C     AIRCRAFT PROFILE.  THE FOLLOWING WILL DEFINE THESE ARRAYS.
C
      VSAZ=AUSC(AZ,A(1,I),VS(1,I),N)
      ALT=CFTM*ALTFM
      VSFM=CMTF*SSPF(ALT)
      VMAZ=AUSC(AZ,A(1,I),VEL(1,I),N)
      IF ((ABS(VMAZ-FMACH)/((VMAZ+FMACH)/2.0)).LE.0.01) VMAZ=FMACH
      ACCAZ=AUSC(AZ,A(1,I),ACCEL(1,I),N)
      CALL PAGE (-6)
      WRITE (6,640) ALTFM
      WRITE (6,650) I,AZ
      WRITE (6,660) VSAZ
      WRITE (6,670) ACCAZ
```

```
      WRITE (6,680) VMAZ,FMACH                                          QUANTA
      FIMAZ=VSAZ*VMAZ                                                   QUANTA
      FMAFT=VSFM*FMACH                                                  QUANTA
      WRITE (6,690) FIMAZ,FMAFT                                         QUANTA
      IF(VMAZ.GT.FMACH) CALL ABORT(WHERE,16,"MACH AT LEVEL OFF CANNOT BE QUANTA
     1 GREATER THAN FINAL MACH")                                        QUANTA
C                                                                       QUANTA
C          EQUATIONS OF MOTION                                          QUANTA
      ACC2=ACCAZ/2.0                                                    QUANTA
      RANAZ=AOSC(AZ,A(1,I),F(1,I),N)                                    QUANTA
      TIMAZ=AOSC(AZ,A(1,I),G(1,I),N)                                    QUANTA
      DO 50 J=1,N                                                       QUANTA
      INDEX=J                                                           QUANTA
      IF ( F(J,I).GE.RANAZ .OR. G(J,I).GE.TIMAZ ) GO TO 60              QUANTA
C           THE FIRST PART OF THE S AND D ARRAYS IS SIMPLY THE FLIGHT   QUANTA
C              PROFILE (F,G) READ IN                                    QUANTA
      D(J,I)=F(J,I)                                                     QUANTA
      S(J,I)=G(J,I)                                                     QUANTA
   50 CONTINUE                                                          QUANTA
   60 D(INDEX,1)=RANAZ                                                  QUANTA
      S(INDEX,1)=TIMAZ                                                  QUANTA
      VINT=FIMAZ                                                        QUANTA
      INDEX=INDEX+1                                                     QUANTA
C                                                                       QUANTA
      IF (VMAZ.EQ.FMACH .OR. AZ.GE.ALTFM .OR. XATI.EQ.0.0) GO TO 80     QUANTA
C           THE FOLLOWING DEFINES S AND D (XATI ADDITIONAL VALUES FOR   QUANTA
C              BOTH ARRAYS) FROM MACH AT LEVEL OFF TO FINAL MACH        QUANTA
      T2=TIMAZ**2                                                       QUANTA
      AT=ACCAZ*TIMAZ                                                    QUANTA
      VINT=FMAFT                                                        QUANTA
      TF=TIMAZ+ (VINT-FTMAZ)/ACCAZ                                      QUANTA
      DT=(TF-TIMAZ)/XATI                                                QUANTA
   70 K=INDEX                                                           QUANTA
      S(K,I)=S(K-1,I)+DT                                                QUANTA
      SK=S(K,I)                                                         QUANTA
      D(K,I)=RANAZ + FTMAZ*(SK-TIMAZ) + ACC2*(SK**2+T2) -AT*SK          QUANTA
      INDEX=INDEX+1                                                     QUANTA
      IF (SK+DT.LT.TF .AND. INDEX.LT.NXAP-1) GO TO 70                   QUANTA
      S(INDEX,I)=TF                                                     QUANTA
      D(INDEX,I)=RANAZ+ FTMAZ*(TF-TIMAZ) + ACC2*(TF**2+T2) -AT*TF       QUANTA
      IF (ABS(TF-S(K,I)).LT.0.001) INDEX=K                              QUANTA
      INDEX=INDEX+1                                                     QUANTA
C                                                                       QUANTA
C           THE S AND D ARRAYS ARE COMPLETED WITH CONSTANT VELOCITY     QUANTA
   80 DO 90 K=INDEX,NXAP                                                QUANTA
      S(K,I)=S(K-1,I)+TINT                                              QUANTA
      D(K,I)=D(K-1,I)+ VINT*TINT                                        QUANTA
   90 CONTINUE                                                          QUANTA
      WRITE (6,700)                                                     QUANTA
      CALL PAGE (10)                                                    QUANTA
```

```
C           CONVERT FEET TO NM AND SECONDS TO MINUTES
       DO 100 K=1,NXAP
       TEMP1=S(K,I)
       TEMP2=D(K,I)
       S(K,I)=S(K,I)/60.0
       D(K,I)=D(K,I)/6080.0
       WRITE (6,710) TEMP2,TEMP1,D(K,I),S(K,I)
       CALL PAGE (1)
  100  CONTINUE
C           COMPUTE OVERPRESSURE AND THERMAL EFFECTS AT A(N,I), MAX ALT
       DO 120 J=1,MTYPES
       HBL=BURST(J)-HTE
       W=YIELD(J)
       TH=CAL(I)
       OP=PSI(I)
       AZ=A(N,I)-HTE
C
       HORF=0.0
       TSA=0.0
       CALL SPARK(2,W,HBL,AZ,HTE,DUM(1),DUM(2),DUM(3),DUM(4),OP,POINTS,
      * XFA,XTPP,XMS,TFA,TTPP,TMS)
       IF (POINTS.EQ.0.) GO TO 110
       HORF=XMS
       TSA=TMS
       IF (POINTS.EQ.2.) GO TO 110
       HORF=XFA
       TSA=TFA
  110  FLRF(I,J)=HORF/6080.0
       FTSA(I,J)=TSA/60.0
C
       SNNEW=.TRUE.
       CALL SNARE(DUM(5),AZ,0.,0.,0.,0.,W,HBL,HTE,RHO,VIS,PZ,0.,0.,0.,
      * GC,JD,GR,TH,HOR,VALID,SNNEW)
       FLRT(I,J)=HOR/6080.0
  120  CONTINUE
C            NEXT AIRCRAFT TYPE
  130  CONTINUE
       RETURN
C
  500  FORMAT (6F10.8)
  510  FORMAT (I5)
  520  FORMAT (3F15.8,2F10.6)
  600  FORMAT (4X,*SUBROUTINE PROCESS*////8X,*AIRCRAFT TYPE *,I2,
      * *  OF *,I2,*  TYPE(S)*,//)
  610  FORMAT (11X,*VULNERABILITY CRITERIA FOR AIRCRAFT TYPE *//
      * 16X,F5.2,5H  PSI,16X,I5,*  CAL/CM*,3H**2//)
  620  FORMAT (11X,*DATA INPUT FOR AIRCRAFT PROFILE*//12X,4HCARD,
      * 10X,6HGROUND,15X,6HFLIGHT,14X,8HAIRCRAFT,13X,8HVELOCITY,11X,
      * 4HMACH,5X,9HLEVEL-OFF/12X,6HNUMBER,9X,5HRANGE,17X,4HTIME,15X,
      * 8HALTITUDE,13X,8HOF SOUND,10X,6HNUMBER,3X,12HACCELERATION/)
```

```
630  FORMAT (12X, I3,4(6X,F15.8),2(6X,F6.3))                                QUANTA   688
640  FORMAT (//16X,*INITIAL ALTITUDE OF MAXIMUM MACH *,F8.1,* FEET*)       QUANTA   689
650  FORMAT(/11X,32HDATA COMPUTED FOR AIRCRAFT TYPE ,I2,* AT A TERMINAL  QUANTA   690
    * ALTITUDE OF *, F8.1,6H  FEET/)                                      QUANTA   691
660  FORMAT (16X,*VELOCITY OF SOUND:*,4X,F8.2,13H  FEET/SECOND/)          QUANTA   692
670  FORMAT (11X,*ACCELERATION COMPONENT:*,4X,F8.3,*  FEET/SEC/SEC*/)     QUANTA   693
680  FORMAT (22X,*MACH NUMBER:*,4X,F8.3,*  TERMINAL*,F6.3/)               QUANTA   694
690  FORMAT(25X,*VELOCITY:*,4X,F8.2,*  FEET/SEC   FINAL*,F8.2,            QUANTA   695
    * *  FEET/SEC*/)                                                      QUANTA   696
700  FORMAT (17X,39HGROUND RANGE (FEET)   FLIGHT TIME (SEC),18X,37HGROU QUANTA   697
    1ND RANGE (NM)   FLIGHT TIME (MIN)/)                                  QUANTA   698
710  FORMAT (20X,E16.8,4X,E16.8,19X,E16.8,4X,E16.8)                        QUANTA   699
     END                                                                  QUANTA   700
```

```
      [illegible] AOIS,[illegible]
     * [illegible]
      DIMENSION [illegible]
     * [illegible]
     * [illegible]
      COMMON /[illegible]/[illegible]
      COMMON /[illegible]/[illegible]
      LOGICAL SPACE
      [illegible]
     * [illegible]
      DO 10 I=1,13
      AOIS(I)=I-1
   10 CONTINUE
      DO 20 I=14,37
      AOIS(I)=(I-11)*5
   20 CONTINUE
C         GENERATE TABLE OF LETHAL AREAS AND DIAMETERS
      DO 100 JT=1,NTYPES
      [illegible]
      DO 100 MT=1,MTYPES
      [illegible]
      CALL PAGE (51)
      WRITE (6,6010) JT,MT
C         SET UP VARIABLES
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
C
      DO 30 I=1,30
      JTYPE=JT
      MISTYP=MT
      AREA(I,JT,MT)=DTAREA(AOIS(I),[illegible]
     * [illegible]3.1415926)
      WRITE (6,6020) AOIS(I),AREA(I,JT,MT)
   30 CONTINUE
      WRITE (6,6030)
      DO 40 I=1,13
      CALL DTLINE (AOIS(I),5.,NTYPES,[illegible])
      DIAM(I,JT,MT)=DBRP
      DIAM(I+15,JT,MT)=DTP
      WRITE (6,6040) AOIS(I),DIAM(I,JT,MT),DIAM(I+15,JT,MT)
   40 CONTINUE
      CALL PAGE(51)
  100 CONTINUE
C
      RETURN
 6010 FORMAT(//21X,10(1H*),* AIRCRAFT TYPE *,I4,* VS. MISSILE TYPE *
     *,I4,10(1H*)//)
```

```
      5020 FORMAT(* WHEN DETONATION IS *,F6.1,* NM FROM CENTROID, LETHAL AREA GOA.IA      751
          * = *,F8.2,* SQUARE NM*)                                            GOA.IA      752
      5030 FORMAT(////)                                                         GOA.IA      753
      5040 FORMAT(* DETONATION IS *,F5.1,* NM FROM BRAKE RELEASE, FALL LINE I GOA.IA      754
          *S*,F5.1,* PLUS *,F5.1,* NM*)                                       GOA.IA      755
           END                                                                GOA.IA      756
```

```
      SUBROUTINE OUTLINE(GNM,DTRNM,NTYPES,NFAP,NSD,S,D,A,F,BRPSIDE,TPSIDE QUANTA   757
     * )                                                                QUANTA   758
      DIMENSION S(NSD,NTYPES),D(NSD,NTYPES),A(NFAP,NTYPES),             QUANTA   759
     * F(NFAP,NTYPES)                                                   QUANTA   760
      EXTERNAL MS,FAS,THRML,TPPI                                        QUANTA   761
      REAL MS                                                           QUANTA   762
      COMMON /OSTIME/HB,HARDTH,HARDOP,W,NAF,I                           QUANTA   763
      COMMON /PASS/D3(3),GALT                                           QUANTA   764
      COMMON /STUFF/G,HTE                                               QUANTA   765
      COMMON /FLICK/ D1(5),GG,D2(8)                                     QUANTA   766
      LEVEL 2,  GNM,S,D,A,F                                             QUANTA   767
      DATA WHERE/6HOUTLINE/                                             QUANTA   768
C                                                                       QUANTA   769
C THIS FUNCTION FINDS THE LETHAL GROUND RANGE ALONG THE A/C FLIGHT      QUANTA   770
C PATH FROM A BLAST GNM NAUTICAL MILES FROM BRAKE RELEASE               QUANTA   771
C LETHAL GROUND RANGE IS RETURNED IN NM. BRPSIDE IS THE DISTANCE IN NM  QUANTA   772
C FROM THE BURST POINT TO THE LETHAL BOUNDARY IN THE DIRECTION OF THE   QUANTA   773
C BRAKE RELEASE POINT.  TPSIDE ISTHE SAME THING IN THE DIRECTION OF     QUANTA   774
C THE TURN POINT.                                                       QUANTA   775
C                                                                       QUANTA   776
C     SET UP VARIABLES                                                  QUANTA   777
      HTE=GALT                                                          QUANTA   778
      G=AMAX1(10.,GNM*6080.)                                            QUANTA   779
      DTP=DTRNM*6080.                                                   QUANTA   780
C                                                                       QUANTA   781
C     FIRST FIND BOUNDARY ON BRP SIDE OF BURST.  TO AVOID DIFFICULTIES  QUANTA   782
C     ENCOUNTERED IN NOC ROUTINES WHEN HAC=0, G=10 FT IS USED INSTEAD OF QUANTA  783
C     G=0 FOR BRP TO GIVE A/C SLIGHT POSITIVE ALTITUDE.                 QUANTA   784
C                                                                       QUANTA   785
      X=0.                                                              QUANTA   786
      HACBR=AOSC(10.,F(1,I),A(1,I),NAF)-HTE                             QUANTA   787
C     CHECK IF BURST IS AT BRP                                          QUANTA   788
      IF (G .LE. 10.) GO TO 100                                         QUANTA   789
C     CHECK MACH SHOCK AT BRP (GG)                                      QUANTA   790
      KGRD=1                                                            QUANTA   791
      CALL BLINIT(HB,W,HTE+HACBR,HTE,KGRD)                              QUANTA   792
      XMS=G                                                             QUANTA   793
C     IF MS OCCURS BEYOND BRP, FIND FREE AIR BOUND                      QUANTA   794
      IF (GG .GE. (G-10.)) GO TO 30                                     QUANTA   795
      SR=SQRT(HACBR**2+(G-10.)**2)                                      QUANTA   796
      OP=TPPKOP(SR,GINT,HINT)                                           QUANTA   797
C     IF OVERPRESSURE AT BRP IS GREATER THAN A/C HARDNESS, WE ARE DONE  QUANTA   798
      IF (OP .GE. HARDOP) GO TO 100                                     QUANTA   799
C     FIND INTERSECTION OF TRIPLE POINT PATH AND FLIGHT PROFILE (XTPPI) QUANTA   800
      CALL REGFALS(TPPI,10.,G-GG,100,50.,0.,XTPPI,IFLAG,F(1,I),A(1,I))  QUANTA   801
      IF(IFLAG.EQ.2) CALL ABORT(WHERE,1,"ROOT CANNOT BE FOUND")         QUANTA   802
      IF(IFLAG.EQ.1) XTPPI=G-GG                                         QUANTA   803
C     FIND MS BOUND BETWEEN BRP AND XTPPI                               QUANTA   804
      CALL REGFALS(MS,10.,XTPPI,100,50.,0.,XMS,IFLAG,F(1,I),A(1,I))     QUANTA   805
      IF(IFLAG.EQ.2) CALL ABORT(WHERE,2,"ROOT CANNOT BE FOUND")         QUANTA   806
```

```
C     FIND FREE AIR SHOCK BOUND
   50 TEST=FAS(10.,F(1,I),A(1,I))
      IF (TEST .GE. 0.) GO TO 100
      CALL REGFALS(FAS,10.,0,100,50.,0.,XFA,IFLAG,F(1,I),A(1,I))
      IF(IFLAG.NE.0) CALL ABORT(WHERE,3,"ROOT CANNOT BE FOUND")
      X=AMIN1(XMS,XFA)
C     BACK UP POINT
      HAC=AUSC(X,F(1,I),A(1,I),NAF)-HTE
      KGRD=1
      CALL BLINIT(HB,W,HAC,HTE,HTE,KGRD)
      SR=SQRT((G-X)**2+(HAC-HB)**2)
      TSA=WFTF(SR)/60.
      TX=PROFLO(X/6080.,U(1,I),S(1,I),NSD)
      T=TX-TSA
      T=AMAX1 (0.,T)
      X=PROFLO(T,S(1,I),D(1,I),NSD)*6080.
      IF (X .EQ. 0.) GOTO 100
C     CHECK THERMAL EFFECT AT OP BOUND
      TEST=THRML(X,F(1,I),A(1,I))
      IF (TEST .LE. 0.) GOTO 100
C     FIND THERMAL BOUND
      CALL REGFALS(THRML,10.,X,100,50.,0.,XTH,IFLAG,F(1,I),A(1,I))
      IF (IFLAG .EQ. 0) X=XTH
      IF (IFLAG .EQ. 1) X=0.
      IF(IFLAG.EQ.2) CALL ABORT(WHERE,4,"ROOT CANNOT BE FOUND")
  100 BRPSIDE=X
C
C     THE BRP SIDE BOUNDARY IS NOW ESTABLISHED.  NEXT FIND TURN POINT
C     SIDE BOUNDARY.
C
C     GET XTPPI ON THIS SIDE
      KGRD=1
      CALL BLINIT(HB,W,HTE,HACBR,HTE,KGRD)
      CALL REGFALS(TPPI,G+G0+1.,G+5.*DTP,100,50.,0.,XTPPI,IFLAG,
     1     F(1,I),A(1,I) )
      IF(IFLAG.EQ.2) CALL ABORT(WHERE,5,"ROOT CANNOT BE FOUND")
C     IF TPP DOES NOT INTERSECT WITH FLIGHT PATH CLOSE ENOUGH TO BURST
C     FIND FREE AIR BOUND
      XMS=0
      IF (IFLAG.EQ.1) GO TO 140
C     FIND MS AT TPPI
      TEST=MS(XTPPI,F(1,I),A(1,I))
C     IF A/C IS ABOVE MACH SHOCK GO TO FAS
      IF (TEST .LE. 0.) GO TO 140
C     FIND MS BOUND
      X2=XTPPI
  130 X1=X2
      X2=X2+XTPPI
      CALL REGFALS(MS,X1,X2,100,50.,0.,XMS,IFLAG,F(1,I),A(1,I))
      IF (IFLAG .EQ. 1) GO TO 130
```

```
      IF(IFLAG.EQ.2) CALL ABORT(WHERE,6,"ROOT CANNOT BE FOUND")        QUANTA   857
C     FIND FAS BOUND                                                   QUANTA   858
  140 X2=G                                                             QUANTA   859
  145 X1=X2                                                            QUANTA   860
      X2=X2+DTP                                                        QUANTA   861
      CALL REGFALS(FAS,X1,X2,100,50.,0.,XFA,IFLAG,F(1,I),A(1,I))       QUANTA   862
      IF (IFLAG .EQ. 1) GO TO 145                                      QUANTA   863
      IF(IFLAG.EQ.2) CALL ABORT(WHERE,7,"ROOT CANNOT BE FOUND")        QUANTA   864
      X=AMAX1(XMS,XFA)                                                 QUANTA   865
C     BACK UP POINT                                                    QUANTA   866
      HAC=AUSC(X,F(1,I),A(1,I),NAF)-HTE                                QUANTA   867
      SR=SQRT((G-X)**2+(HAC-HE)**2)                                    QUANTA   868
      KGRD=1                                                           QUANTA   869
      CALL BLINIT(HB,W,HAC+HTE,HTE,KGRD)                               QUANTA   870
      TSA=WFTF(SR)/60.                                                 QUANTA   871
      TX=PROFLD(X/6080.,D(1,I),S(1,I),NSD)                             QUANTA   872
      T=TX-TSA                                                         QUANTA   873
      T=AMAX1 (0.,T)                                                   QUANTA   874
      X=PROFLD(T,S(1,I),D(1,I),NSD)*6080.                              QUANTA   875
      X=AMAX1(X,G)                                                     QUANTA   876
C     CHECK THERMAL EFFECT AT X                                        QUANTA   877
      TEST=THRML(X,F(1,I),A(1,I))                                      QUANTA   878
      IF (TEST .LE. 0.) GOTO 200                                       QUANTA   879
C     FIND THERMAL BOUND                                               QUANTA   880
      X2=X                                                             QUANTA   881
  160 X1=X2                                                            QUANTA   882
      X2=X2+DTP                                                        QUANTA   883
      CALL REGFALS(THRML,X1,X2,100,50.,0.,XTH,IFLAG,F(1,I),A(1,I))     QUANTA   884
      IF(IFLAG.EQ.2) CALL ABORT(WHERE,8,"ROOT CANNOT BE FOUND")        QUANTA   885
      IF (IFLAG .EQ. 1) GOTO 160                                       QUANTA   886
      X=XTH                                                            QUANTA   887
  200 TPSIDE=X                                                         QUANTA   888
C                                                                      QUANTA   889
C     BOTH BOUNDARIES ARE KNOWN.                                       QUANTA   890
C     LETHAL DISTANCE IS DISTANCE FROM BRP SIDE TO TP SIDE.            QUANTA   891
C     MAKE BRPSIDE AND TPSIDE RELATIVE TO BURST POINT                  QUANTA   892
      BRPSIDE=GNM-BRPSIDE/6080.                                        QUANTA   893
      TPSIDE=TPSIDE/6080.-GNM                                          QUANTA   894
      RETURN                                                           QUANTA   895
      END                                                              QUANTA   896
```

```
      SUBROUTINE REGFALS(FN,AOR,BOR,NTOL,XTOL,FTOL,W,IFLAG,FA,AA)
C   THIS SUBROUTINE FINDS THE REAL ROOTS OF A FUNCTION GIVEN AS
C   AN EXTERNAL.  THE USER MUST GIVE BOUNDRY POINTS THAT LIE ON
C   EITHER SIDE OF THE ROOT.  THESE POINTS ARE 'A' AND 'B'.
C   OTHER VARIABLES ARE:
C              FN - THE FUNCTION GIVEN BY THE USER
C              AOR - THE LEFTMOST BOUND ON THE ROOT
C              BOR - THE RIGHTMOST BOUND ON THE ROOT
C              NTOL - THE MAXIMUM NUMBER OF ITERATIONS ALLOWED BY USER
C              XTOL - THE TOLERANCE SPECIFIED BY USER FOR THE
C                     DIFFERANCE BETWEEN TWO ITERATIONS
C              FTOL - THE MAXIMUM ERROR IN THE FUNCTION AT THE ROOT
C              W - THE VALUE OF THE ROOT
C              IFLAG - A FLAG TO TELL THE USER HOW THE PROGRAM RAN
C                      0. THE ROOT WAS FOUND TO WITHIN THE USERS
C                         TOLERANCE.
C                      1. THE ENDPOINTS DO NOT BOUND A REAL ROOT
C                      2. THE ROUTINE DID NOT CONVERGE TO WITHIN
C                         THE USERS TOLERANCES.
      LEVEL 2,FA,AA
      A=AOR
      B=BOR
      F=FN(A,FA,AA)
      G=FN(B,FA,AA)
C
C   CHECK TO SEE IF THE ENDPOINTS BOUND A ROOT.
      SF=SIGN(1.,F)
      IF((SF*G).LE.0.)GO TO 10
      IFLAG=1
      GO TO 50
   10 IFLAG=0
      W=A
      FW=F
      DO 30 N=1,NTOL
C
C   CHECK FOR SMALL ENOUGH INTERVAL BETWEEN ROOTS
      IF(ABS((B-A)/2.).LE.XTOL)GO TO 50
C
C   CHECK FOR SMALL ENOUGH FUNCTION VALUES
      IF(ABS(FW).LE.FTOL)GO TO 50
      W=((F*B)-(G*A))/(F-G)
      SW=SIGN(1.,FW)
      FW=FN(W,FA,AA)
      IF((SF*FW).LT.0.)GO TO 20
C
C   CALCULATE NEW A, F, AND CUT G IN HALF.
      A=W
      F=FW
      IF ((SW*FW).GT.0.) G=G/2.
      GO TO 30
```

```
C
C   CALCULATE NEW B, G, AND CUT F IN HALF.
    20 B=W
       G=FW
       IF ((SW*FW).GT.0.) F=F/2.
    30 CONTINUE
C
C   SET IFLAG TO 2 IF THE PROGRAM DIDN'T CONVERGE WITHIN LIMITS
       IFLAG=2
    50 RETURN
       END
```

```
      FUNCTION TPPI(X,F,A)                                             QUANTA   958
C   TRIPLE POINT PATH INTERCEPT........USED IN OUTLINE                 QUANTA   959
      COMMON /STUFF/G                                                  QUANTA   960
      COMMON /FLICK/U1(5),G0,CTP                                       QUANTA   961
      COMMON /CSTIME/U2(4),NAF                                         QUANTA   962
      LEVEL 2,F,A                                                      QUANTA   963
      H1=G0*CTP*(ABS(G-X)/G0-1.)**1.6                                  QUANTA   964
      H2=AUSC(X,F,A,NAF)                                               QUANTA   965
      TPPI=H1-H2                                                       QUANTA   966
      RETURN                                                           QUANTA   967
      END                                                              QUANTA   968
```

```
      REAL FUNCTION MS(X,F,A)                                             QUANTA   969
C  MACH SHOCK............USED IN OTLINE                                  QUANTA   970
      COMMON /STUFF/G,HTE                                                 QUANTA   971
      COMMON /OSTIME/HB,HARDTH,HARDOP,W,NAF                               QUANTA   972
      LEVEL 2,F,A                                                         QUANTA   973
      HAC=AOSC(X,F,A,NAF)-HTE                                             QUANTA   974
      KORD=1                                                              QUANTA   975
      CALL BLINIT(HB,W,HAC+HTE,HTE,KORD)                                  QUANTA   976
      SR=SQRT((G-X)**2+HAC**2)                                            QUANTA   977
      MS=TPPROP(SR,GINT,HINT)-HARDOP                                      QUANTA   978
      RETURN                                                              QUANTA   979
      END                                                                 QUANTA   980
```

```
      FUNCTION FAS(X,F,A)
C  FREE AIR SHOCK............USED IN DTLINE
      COMMON /STUFF/Q,HTE
      COMMON /CSTIME/HB,HARDTH,HARDOP,W,NAF
      LEVEL 2,F,A
      HAC=AOSC(X,F,A,NAF)-HTE
      KGRD=1
      CALL BLINIT(HB,W,HAC,HTE,HTE,KGRD)
      SR=SQRT((Q-X)**2+(HAC-HB)**2)
      FAS=WFPROP(SR)-HARDOP
      RETURN
      END
```

```
      FUNCTION THRML(X,F,A)
C  THERMAL EFFECTS..............USED IN OTLINE
      COMMON /STUFF/Q,HTE
      COMMON /CSTIME/HB,HARDTH,HARDOP,W,NAF
      COMMON /PASS/RHO,VIS,PZ
      LEVEL 2,F,A
      LOGICAL SNEW,VALID
      HAC=ABSC(X,F,A,NAF)-HTE
      CALL SHARE(ABS(G-X),HAC,0.,0.,0.,0.,W,HB,HTE,RHO,VIS,PZ,0.,0.,0.,
     1      CAL,GD,GR,-1.,DUM,VALID,SNEW)
      THRML=CAL-HARDTH
      RETURN
      END
```

```
      FUNCTION DTAREA(GNM,DTPNM,NTYPES,MTYPES,NPAP,NSD,S,D,A,F,
     * FLRP,FTSA,FLRT,SNNEW,THETA)
      DIMENSION S(NSD,NTYPES),D(NSD,NTYPES),A(NPAP,NTYPES),F(NPAP,NTYPES
     * ),FLRP(NTYPES,MTYPES),FTSA(NTYPES,MTYPES),FLRT(NTYPES,MTYPES)
      COMMON /DSTIME/HB,HARDTH,HARDOP,W,NAF,ITYPE,JTYPE
      COMMON /PASS/RHO,VIS,PZ,HTE
      LEVEL 2, GNM,S,D,A,F,FLRP,FTSA,FLRT
      LOGICAL VALID,SNNEW
C
C   THIS FUNCTION CALCULATES THE LETHAL AREA FOR A BURST LOCATED GNM NM
C   FROM THE CENTROID.   DTPNM IS THE DISTANCE TO THE TURN POINT IN NM.
C   THETA IS HALF OF THE FLYOUT SECTOR MEASUREMENT IN RADIANS.  THE AREA
C   RETURNED IS IN NM**2.
C
      XBRST=GNM+DTPNM
C    CHECK IF PAST LEVEL OFF DISTANCE
      IF (XBRST*6080. .LT. F(NAF,ITYPE)) GO TO 10
C    IF PAST USE STORED DATA
      ROP=FLRP(ITYPE,JTYPE)
      TSA=FTSA(ITYPE,JTYPE)
      RTH=FLRT(ITYPE,JTYPE)
      GOTO 40
C    NEED TO CALCULATE DATA
C    SET UP VARIABLES
   10 HAC=AOSC(XBRST*6080.,F(1,ITYPE),A(1,ITYPE),NAF)-HTE
C    GET HORIZONTAL RANGE TO OP BOUNDARY (ROP) AND TSA AT BOUND
      ROP=0.
      TSA=0.
      CALL SPARK(2,W,HB,HAC,HTE,D1,D2,D3,D4,HARDOP,POINTS,XFA,XTPP,XMS,
     1    TFA,TTPP,TMS)
      IF (POINTS .EQ. 0.) GO TO 20
      ROP=XMS
      TSA=TMS
      IF(POINTS .EQ. 2.) GO TO 20
      ROP=XFA
      TSA=TFA
C    CHANGE FT TO NM AND SEC TO MIN
   20 ROP=ROP/6080.
      TSA=TSA/60.
C    FIND RANGE TO THERMAL BOUNDARY (RTH)
      CALL SNAKE(0.,HAC,0.,0.,0.,0.,W,HB,HTE,RHO,VIS,PZ,0.,0.,0.,GC,GD,
     1    GR,HARDTH,RTH,VALID,SNNEW)
C    CHANGE FT TO NM
      RTH=RTH/6080.
C
C    WE NOW NEED AN ESTIMATE OF THE AVERAGE DISTANCE TRAVELED BY THE
C    RECEIVER BETWEEN DETONATION AND SHOCK ARRIVAL. TO BE USED IN
C    BACKING UP THE OVERPRESSURE CONTOUR.  WE WILL GET THIS ESTIMATE
C    BY WORKING BACKWARD FROM THE DEGNATION POINT.
C
```

```
 40     TTB=PROFLU(XBRST,D(1,ITYPE),S(1,ITYPE),NSD)              QUANTA    1056
        T=TTB-TSA                                                QUANTA    1057
        T=AMAX1 (0.,T)                                           QUANTA    1058
        DIS=XBRST-PROFLU(T,S(1,ITYPE),D(1,ITYPE),NSD)            QUANTA    1059
 C     NOW FIND THE TOTAL LETHAL AREA                            QUANTA    1060
        DTAREA=THOPA(ROP,GNM,RTH,DIS,THETA)                      QUANTA    1061
        RETURN                                                   QUANTA    1062
        END                                                      QUANTA    1063
```

PAGE 34

```
      FUNCTION THOPA(ROP,G,RTH,PROJ,THTA)                               QUANTA  1064
      LEVEL 2,G                                                         QUANTA  1065
C                                                                       QUANTA  1066
C   PURPOSE                                                             QUANTA  1067
C        TO COMPUTE THE AREA OF THE FIGURE FORMED BY THE UNION OF THE   QUANTA  1068
C        OVER PRESSURE "SQUISH" AND THE THERMAL CIRCLE INTERSECTID WITH QUANTA  1069
C        THE SPACE TO THE SIDE OF THE LINE OF FLIGHT WITH DECREASING    QUANTA  1070
C        THETA.                                                         QUANTA  1071
C                                                                       QUANTA  1072
      LOGICAL BEXIST,CEXIST                                             QUANTA  1073
      DATA NOINC /500/                                                  QUANTA  1074
      DATA EPSLON/-1.0E-8/                                              QUANTA  1075
C        INITILIZE.                                                     QUANTA  1076
C                                                                       QUANTA  1077
      THOPA=0.0                                                         QUANTA  1078
      THETA=ABS(THTA)                                                   QUANTA  1079
      ROPSQ=ROP*ROP                                                     QUANTA  1080
      RTHSQ=RTH*RTH                                                     QUANTA  1081
      SI=0.                                                             QUANTA  1082
      R2OLD=AMAX1(G+ROP-PROJ,G+RTH)                                     QUANTA  1083
      R1OLD=AMIN1(G-ROP-PROJ,G-RTH)                                     QUANTA  1084
      AOLD=G-ROP-PROJ                                                   QUANTA  1085
      BOLD=G+ROP-PROJ                                                   QUANTA  1086
      COLD=G-RTH                                                        QUANTA  1087
      DOLD=G+RTH                                                        QUANTA  1088
C        DETERMINE THE LARGEST ANGLE AND THE INCREMENTIAL ANGLE.        QUANTA  1089
C                                                                       QUANTA  1090
      IF((G.GE.RTH).AND.(G.GE.ROP)) THETA=AMIN1(THETA,AMAX1(ASIN(ROP/G),QUANTA  1091
     1   ASIN(RTH/G)))                                                  QUANTA  1092
      DTHETA=THETA /FLOAT(NOINC)                                        QUANTA  1093
C                                                                       QUANTA  1094
C        STARTING AT ZERO INCREMENT SI UP TO THETA COMPUTING AREAS AT   QUANTA  1095
C        EACH INCREMENT                                                 QUANTA  1096
C                                                                       QUANTA  1097
      DO 40 J=1,NOINC                                                   QUANTA  1098
C        SET UP THE ANGLE.                                              QUANTA  1099
C                                                                       QUANTA  1100
      SI=FLOAT(J)*DTHETA                                                QUANTA  1101
      COSSI=COS(SI)*G                                                   QUANTA  1102
      SINSI=SIN(SI)*G                                                   QUANTA  1103
      SINSQ=SINSI*SINSI                                                 QUANTA  1104
C        COMPUTE THE POINTS OF INTERSECTION.                            QUANTA  1105
      A=1.0E295                                                         QUANTA  1106
      B=-1.0E295                                                        QUANTA  1107
      BEXIST=.FALSE.                                                    QUANTA  1108
C                                                                       QUANTA  1109
      DESC=ROPSQ-SINSQ                                                  QUANTA  1110
      IF(DESC.GT.EPSLON.AND.DESC.LT.0.0) DESC = 0.0                     QUANTA  1111
      IF(DESC.LT.0.0) GO TO 10                                          QUANTA  1112
      DESC=SQRT(DESC)                                                   QUANTA  1113
```

```
      A=COSSI-PROJ-DESC                                          QUANTA   1114
      B=COSSI-PROJ+DESC                                          QUANTA   1115
      A=AMAX1(A,0.0)                                             QUANTA   1116
      B=AMAX1(B,0.0)                                             QUANTA   1117
      BEXIST=.TRUE.                                              QUANTA   1118
C                                                                QUANTA   1119
   10 C=1.0E295                                                  QUANTA   1120
      D=-1.0E295                                                 QUANTA   1121
      CEXIST=.FALSE.                                             QUANTA   1122
C                                                                QUANTA   1123
      DESC=RTHSQ-SINSQ                                           QUANTA   1124
      IF(DESC.GT.EPSLON.AND.DESC.LT.0.0) DESC = 0.0              QUANTA   1125
      IF(DESC.LT.0.0) GOTO 20                                    QUANTA   1126
      C=COSSI-SQRT(DESC)                                         QUANTA   1127
      D=COSSI+SQRT(DESC)                                         QUANTA   1128
      C=AMAX1(C,0.0)                                             QUANTA   1129
      D=AMAX1(D,0.0)                                             QUANTA   1130
C                                                                QUANTA   1131
C        COMPUTE THE AREAS.                                      QUANTA   1132
C                                                                QUANTA   1133
20    IF(.NOT.(CEXIST.OR.BEXIST)) GO TO 50                       QUANTA   1134
      IF((B.LE.C).AND.BEXIST.AND.CEXIST) GO TO 30                QUANTA   1135
C        ONE AREA                                                QUANTA   1136
C                                                                QUANTA   1137
      R1=AMIN1(A,C)                                              QUANTA   1138
      R2=AMAX1(B,D)                                              QUANTA   1139
      THOPA=THOPA+(R2*R2OLD-R1*R1OLD)*DTHETA                     QUANTA   1140
      R1OLD=R1                                                   QUANTA   1141
      R2OLD=R2                                                   QUANTA   1142
      GO TO 40                                                   QUANTA   1143
C        TWO AREAS                                               QUANTA   1144
C                                                                QUANTA   1145
   30 THOPA=THOPA+((B*BOLD-A*AOLD)+(D*DOLD-C*COLD))*DTHETA       QUANTA   1146
      AOLD=A                                                     QUANTA   1147
      BOLD=B                                                     QUANTA   1148
      COLD=C                                                     QUANTA   1149
      DOLD=D                                                     QUANTA   1150
C                                                                QUANTA   1151
   40 CONTINUE                                                   QUANTA   1152
50    RETURN                                                     QUANTA   1153
      END                                                        QUANTA   1154
```

```
      SUBROUTINE ZERO (NTGTS,NSUBS)                                   QUANTA  1155
C                                                                     QUANTA  1156
C     INITIALIZES FLAGS TO ZERO                                       QUANTA  1157
C                                                                     QUANTA  1158
      COMMON/RNGFLG/BASENO(47),NBITS                                  QUANTA  1159
      DATA NOBASE/47/                                                 QUANTA  1160
      DATA NBITS/60/                                                  QUANTA  1161
C                                                                     QUANTA  1162
C        CHECK FOR TOO MANY BASES                                     QUANTA  1163
      IF(NTGTS.GT.NOBASE) STOP "IN ZERO,    NTGTS .GT. NOBASE"        QUANTA  1164
C        CHECK FOR TOO MANY SUBS                                      QUANTA  1165
      IF(NSUBS.GT.NBITS) STOP"IN ZERO,    NSUBS .GT. NBITS"           QUANTA  1166
      DO 10 I=1,NOBASE                                                QUANTA  1167
      BASENO(I)=0.                                                    QUANTA  1168
   10 CONTINUE                                                        QUANTA  1169
C                                                                     QUANTA  1170
      RETURN                                                          QUANTA  1171
      END                                                             QUANTA  1172
```

```
      SUBROUTINE SRVMAT(SRVLN,SRVPC,BRTIME,BRT,NRWAYS,ISEQ,DELTAC         QUANTA  1173
     * ,TGTLAT,TGTLNG,SUBLAT,SUBLNG,MTYPE,REL,RVS,RNGMIN,RNGMAX,MPROF,    QUANTA  1174
     * FMRNG,FMTIME,NMPS,DELTM,AREAL,DIAM,DTCENT,S,D,RAN,GPTS,NWPNS,      QUANTA  1175
     * MXNAPB,NTYPES,MTYPES,NTGTS,NACT,MXRWAY,NSUBS,MXMPR,NSD,VAL)        QUANTA  1176
      DIMENSION SRVLN(NWPNS,MXNAPB),SRVPC(NWPNS,MXNAPB),BRTIME(NTYPES),  QUANTA  1177
     * BRT(MXNAPB),NRWAYS(NTGTS),ISEQ(NACT),DELTAC(MXRWAY,NTYPES,NTYPES) QUANTA  1178
     * ,TGTLAT(NTGTS),TGTLNG(NTGTS),SUBLAT(NSUBS),SUBLNG(NSUBS),          QUANTA  1179
     * MTYPE(NSUBS),REL(MTYPES),RVS(MTYPES),RNGMIN(MTYPES),RNGMAX         QUANTA  1180
     * (MTYPES),MPROF(MTYPES),FMRNG(MXMPR,MTYPES),FMTIME(MXMPR,MTYPES),  QUANTA  1181
     * NMPS(NSUBS),DELTM(MTYPES),AREAL(30,NTYPES,MTYPES),                QUANTA  1182
     * DIAM(30,NTYPES,MTYPES),DTCENT(NTYPES),GPTS(30),RAN(MXNAPB),       QUANTA  1183
     * S(NSD,NTYPES),D(NSD,NTYPES),VAL(NTGTS,NWPNS)                      QUANTA  1184
      LEVEL 2,SRVLN,SRVPC,BRTIME,BRT,NRWAYS,ISEQ,DELTAC,TGTLAT,          QUANTA  1185
     * TGTLNG,SUBLAT,SUBLNG,MTYPE,REL,RVS,RNGMAX,RNGMIN,MPROF,FMRNG,     QUANTA  1186
     * FMTIME,NMPS,DELTM,AREAL,DIAM,DTCENT,S,D,GPTS,VAL,RAN              QUANTA  1187
C                                                                         QUANTA  1188
      DATA PI/3.14159265/                                                 QUANTA  1189
      CALL PAGE (0)                                                       QUANTA  1190
      LAST=0                                                              QUANTA  1191
      DO 140 I=1,NTGTS                                                    QUANTA  1192
C                                                                         QUANTA  1193
C     COMPUTE SURVIVABILITY OF EACH AIRCRAFT ON BASE FROM ALL WEAPONS     QUANTA  1194
C                                                                         QUANTA  1195
      IFRST=LAST+1                                                        QUANTA  1196
      DO 10 JT=1,NTYPES                                                   QUANTA  1197
   10 LAST=LAST+INT(VAL(I,JT)+.1)                                        QUANTA  1198
      NAC=LAST-IFRST+1                                                    QUANTA  1199
C          COMPUTE THE BRAKE RELEASE TIMES(BRT) OF ALL AIRCRAFT           QUANTA  1200
C          ON BASE I.                                                     QUANTA  1201
      NWY=NRWAYS(I)                                                       QUANTA  1202
      ITYPE=ISEQ(IFRST)                                                   QUANTA  1203
      BRT(1)=BRTIME(ITYPE)                                                QUANTA  1204
C        IF THERE IS ONLY ONE PLANE ON BASE, DON'T GO THRU LOOP           QUANTA  1205
      IF(NAC.EQ.1) GO TO 30                                               QUANTA  1206
      IAC=IFRST                                                           QUANTA  1207
      DO 20 NI=2,NAC                                                      QUANTA  1208
      IAC=IAC+1                                                           QUANTA  1209
      ITYP1=ITYPE                                                         QUANTA  1210
      ITYPE=ISEQ(IAC)                                                     QUANTA  1211
   20 BRT(NI)=BRT(NI-1)+DELTAC(NWY,ITYP1,ITYPE)                          QUANTA  1212
   30 CONTINUE                                                            QUANTA  1213
C          COMPUTE THE AVERAGE DISTANCE TO THE TURN POINT                 QUANTA  1214
      SUM=0.                                                              QUANTA  1215
      DO 35 IAC=IFRST,LAST                                                QUANTA  1216
      JT=ISEQ(IAC)                                                        QUANTA  1217
      SUM=SUM+DTCENT(JT)                                                  QUANTA  1218
   35 CONTINUE                                                            QUANTA  1219
      AVDTC=SUM/FLOAT(NAC)                                                QUANTA  1220
C                                                                         QUANTA  1221
C     BEGIN THE OUTER LOOP FOR SUBMARINE LOCATIONS                        QUANTA  1222
```

```
      C                                                                  QUANTA  1223
            ISALVO=0                                                     QUANTA  1224
            DO 130 KSUB=1,NSUBS                                          QUANTA  1225
            DISTT=DIST(TGTLAT(I),TGTLNG(I),SUBLAT(KSUB),SUBLNG(KSUB))    QUANTA  1226
            MISTYP=MTYPE(KSUB)                                           QUANTA  1227
            NMIS=NMPS(KSUB)                                              QUANTA  1228
            CALL PAGE (-2)                                               QUANTA  1229
            WRITE (6,610) I,KSUB,DISTT,AVDTC                             QUANTA  1230
      C          IF THE DISTANCE (BETWEEN SUB KSUB AND BASE I) IS OUT OF QUANTA  1231
      C          RANGE, MAKE THE SURVIVABILITY OF THE AIRCRAFT 1.        QUANTA  1232
            IF(DISTT.GE.RNGMIN(MISTYP).AND.DISTT.LE.RNGMAX(MISTYP)) GO TO 50 QUANTA  1233
            CALL SET(I,KSUB)                                             QUANTA  1234
            LST=ISALVO+NMIS                                              QUANTA  1235
            ISALVO=ISALVO+1                                              QUANTA  1236
            DO 40 NI=1,NAC                                               QUANTA  1237
            DO 40 K=ISALVO,LST                                           QUANTA  1238
            SRVPC(K,NI)=..                                               QUANTA  1239
            SRVLN(K,NI)=1.                                               QUANTA  1240
       40   CONTINUE                                                     QUANTA  1241
            ISALVO=LST                                                   QUANTA  1242
      C        CONTINUE TO THE NEXT SUB LOCATION                         QUANTA  1243
            GO TO 130                                                    QUANTA  1244
      C                                                                  QUANTA  1245
      C          COMPUTE THE FIRST WEAPON'S TIME OF ARRIVAL(TOAR) IN MINUTES QUANTA  1246
       50   MPROFX=MPROF(MISTYP)                                         QUANTA  1247
            TOAR = AOSC(DISTT,FMRNG(1,MISTYP),FMTIME(1,MISTYP),MPROFX)   QUANTA  1248
      C                                                                  QUANTA  1249
      C     START THE INNER LOOP FOR MISSILE SALVOS AT KSUB SUB LOC.     QUANTA  1250
      C                                                                  QUANTA  1251
            RVSM=RVS(MISTYP)                                             QUANTA  1252
            RELM=REL(MISTYP)                                             QUANTA  1253
            DO 120 KSAL=1,NMIS                                           QUANTA  1254
            ISALVO=ISALVO+1                                              QUANTA  1255
            TOA=TOAR+FLOAT(KSAL-1)*DELTM(MISTYP)                         QUANTA  1256
      C          COMPUTE THE GROUND RANGE, FROM BRAKE RELEASE, OF EACH   QUANTA  1257
      C          AIRCRAFT(RAN) AT THE TIME OF WEAPON ARRIVAL             QUANTA  1258
            NI=0                                                         QUANTA  1259
            RFA=0.                                                       QUANTA  1260
            DO 60 IAC=IFRST,LAST                                         QUANTA  1261
            NI=NI+1                                                      QUANTA  1262
            JT=ISEQ(IAC)                                                 QUANTA  1263
            RTOA=TOA-BRT(NI)                                             QUANTA  1264
            RA = PROFLU(RTOA,S(1,JT),D(1,JT),NSD)                        QUANTA  1265
            RAN(NI)=RA                                                   QUANTA  1266
            RFA=AMAX1(RA,RFA)                                            QUANTA  1267
       60   CONTINUE                                                     QUANTA  1268
      C          RANGE OF THE FARTHEST AIRCRAFT IF RELATIVE TO THE TURN PT QUANTA  1269
             RFA=RFA-AVDTC                                               QUANTA  1270
            IF (KSAL.NE.1) GO TO 65                                      QUANTA  1271
            WRITE(6,620)TOA,ISALVO,(RAN(IAC),IAC=1,NI)                   QUANTA  1272
```

```
      CALL PAGE (2*(NAC-1)/10)                                          QUANTA  1273
   65 CONTINUE                                                          QUANTA  1274
C                                                                       QUANTA  1275
C     COMPUTE THE SURVIVABILITY FOR EACH AIRCRAFT ON BASE I.            QUANTA  1276
C     THE SURVIVABILITY MATRICES FOR A WEAPON PLACED:                   QUANTA  1277
C      CASE 1-- ON THE RUNWAY (SRVLN)-ALL RVS WILL BE EVENLY DISTRI-    QUANTA  1278
C               BUTED ON THE RUNWAY (FROM THE BRAKE RELEASE POINT TO    QUANTA  1279
C               THE TURN POINT).                                        QUANTA  1280
C      CASE 2-- PAST THE CENTROID (SRVPC)-RV'S WILL BE UNIFORMLY DIS-   QUANTA  1281
C               TRIBUTED IN THE AREA OF THE CIRCLE SWEPT OUT BY THE     QUANTA  1282
C               FARTHEST AIRCRAFT MINUS THE RUNWAY.                     QUANTA  1283
      JTLN=-1                                                           QUANTA  1284
      JTPC=-1                                                           QUANTA  1285
      NI=0                                                              QUANTA  1286
      DO 120 IAC=IFRST,LAST                                             QUANTA  1287
      NI=NI+1                                                           QUANTA  1288
      JT=ISEQ(IAC)                                                      QUANTA  1289
C          COMPUTE RADIUS OF A 180 TURN FOR THIS TYPE OF AC             QUANTA  1290
      TTP=PROFLD(AVDTC,D(1,JT),S(1,JT),NSD)+1.                          QUANTA  1291
      DTP=PROFLD(TTP,S(1,JT),D(1,JT),NSD)-AVDTC                         QUANTA  1292
      TRN=2.*DTP/PI                                                     QUANTA  1293
      IF(RAN(NI).GT.AVDTC+TRN) GO TO 90                                 QUANTA  1294
      IF(JT.EQ.JTLN) GO TO 110                                          QUANTA  1295
      JTLN=JT                                                           QUANTA  1296
C        THIS PLANE IS VULNERABLE TO A LINE ATTACK.                     QUANTA  1297
C        CONSIDER A "TOTAL" LETHAL DIAMETER FROM B.R. TO AVDTC          QUANTA  1298
      SUM=0.                                                            QUANTA  1299
      XINT=AVDTC/RVSM                                                   QUANTA  1300
      NINT=RVSM                                                         QUANTA  1301
      X=-XINT*.5                                                        QUANTA  1302
      TPS= -AVDTC                                                       QUANTA  1303
      DO 80 IN=1,NINT                                                   QUANTA  1304
      X=X+XINT                                                          QUANTA  1305
      BR = AUSC(X,QPTS,DIAM(1,JT,MISTYP),13)                            QUANTA  1306
      TP = AUSC(X,QPTS,DIAM(16,JT,MISTYP),13)                           QUANTA  1307
      SUM=SUM+BR+TP                                                     QUANTA  1308
      SUM=SUM- AMAX1(0.,TPS+BR-XINT)                                    QUANTA  1309
      TPS=TP                                                            QUANTA  1310
   80 CONTINUE                                                          QUANTA  1311
      SUM=SUM -AMAX1(0.,X+TP-AVDTC-TRN)                                 QUANTA  1312
C      FOR CASE 1- THE SURV. DEPENDS ON THE RELIABILITY OF THE MISSILE, QUANTA  1313
C                  THIS TOTAL LETHAL DIAMETER, AND THE LENGTH OF THE    QUANTA  1314
C                  "RUNWAY".                                            QUANTA  1315
C      FOR CASE 2- SURV=1.                                              QUANTA  1316
      PK=SUM/(AVDTC+TRN)                                                QUANTA  1317
      PK=AMIN1(1.,PK)                                                   QUANTA  1318
      SRVLN(ISALVO,NI)=1.-PK*RELM                                       QUANTA  1319
      SRVPC(ISALVO,NI)=1.                                               QUANTA  1320
C        CONTINUE TO THE NEXT AIRCRAFT                                  QUANTA  1321
      GO TO 120                                                         QUANTA  1322
```

```
      C                                                                   QUANTA  1323
   90 IF(JT.EQ.JTPC) GO TO 110                                            QUANTA  1324
      JTPC=JT                                                             QUANTA  1325
      C     THIS PLANE IS VULNERALBE TO A BARRAGE ATTACK.                 QUANTA  1326
      C     SINCE THE LETHAL AREAS DIMINISH AS THEY MOVE OUT FROM THE     QUANTA  1327
      C     CENTROID, A WEIGHTED AVERAGE AREA WILL BE USED.               QUANTA  1328
      IN=2                                                                QUANTA  1329
      SUM=0.0                                                             QUANTA  1330
  100 DELTA=AREAL(IN-1,JT,MISTYP)-AREAL(IN,JT,MISTYP)                     QUANTA  1331
      SUM=SUM+DELTA*((QPTS(IN)+QPTS(IN-1))**2)                            QUANTA  1332
      IN=IN+1                                                             QUANTA  1333
      IF(IN.LE.30 .AND. QPTS(IN).LT.RFA) GO TO 100                        QUANTA  1334
      RHO=(RFA+QPTS(IN-1))**2                                             QUANTA  1335
      T1=AREAL(IN-1,JT,MISTYP)*RHO                                        QUANTA  1336
      AR=AOSC(RFA,QPTS,AREAL(1,JT,MISTYP),30)                             QUANTA  1337
      T2=AR*(4.0*(RFA**2)-RHO)                                            QUANTA  1338
      SUM=(SUM+T1+T2)/(4.0*RFA**2)                                        QUANTA  1339
      WRD=SQRT(SUM/PI)                                                    QUANTA  1340
      BP=(RFA+TRN)*0.5                                                    QUANTA  1341
      DTC=AVDTC                                                           QUANTA  1342
      WAREA = CIRINT(RFA,WRD,BP) - CIRINT(TRN,WRD,BP)                     QUANTA  1343
      IF(RFA.LE.DTC) TA = (RFA**2)*(PI-ASIN(TRN/RFA)) - PI*(TRN**2)/2.    QUANTA  1344
     * - TRN*(SQRT(RFA**2-TRN**2))                                        QUANTA  1345
      IF(RFA.GT.DTC .AND. RFA.LT.DTC+TRN) TA = RFA**2*PI - 2.*TRN*DTC     QUANTA  1346
     * - CIRINT(RFA,TRN,DTC)                                              QUANTA  1347
      IF(RFA.GE.DTC+TRN) TA= PI*(RFA**2-TRN**2) - 2.*TRN*DTC              QUANTA  1348
      C    FOR CASE 1- SURV=1                                             QUANTA  1349
      C    FOR CASE 2- THE SURV. DEPENDS ON THE RELIABILITY OF THE MISSILE, QUANTA  1350
      C              THE WEIGHTED AREA, AND THE AREA THAT THE PLANE IS IN. QUANTA  1351
      SRVLN(ISALVO,NI)=1.                                                 QUANTA  1352
      PK = WAREA*RVSM/TA                                                  QUANTA  1353
      PK=AMIN1(1.,PK)                                                     QUANTA  1354
      SRVPC(ISALVO,NI) = 1.0 - PK*RELM                                    QUANTA  1355
      C     CONTINUE TO THE NEXT AIRCRAFT                                 QUANTA  1356
      GO TO 120                                                           QUANTA  1357
      C                                                                   QUANTA  1358
      C     PLANES OF THE SAME TYPE, HAVING THE SAME POSITION (FALL UNDER QUANTA  1359
      C     THE SAME ATTACK POSITIONS) HAVE THE SAME SURVIVAL.            QUANTA  1360
  110 CONTINUE                                                            QUANTA  1361
      SRVLN(ISALVO,NI)=SRVLN(ISALVO,NI-1)                                 QUANTA  1362
      SRVPC(ISALVO,NI)=SRVPC(ISALVO,NI-1)                                 QUANTA  1363
  120 CONTINUE                                                            QUANTA  1364
  130 CONTINUE                                                            QUANTA  1365
      IN=2*(I-1)                                                          QUANTA  1366
      CALL WRITMS(4,SRVLN(1,1),NWPNS*MXNAPB,IN+1)                         QUANTA  1367
      CALL WRITMS(4,SRVPC(1,1),NWPNS*MXNAPB,IN+2)                         QUANTA  1368
  140 CONTINUE                                                            QUANTA  1369
      RETURN                                                              QUANTA  1370
  610 FORMAT (/* THE DISTANCE TO TARGET *,I3,* FROM SUB *,I2,* IS *,      QUANTA  1371
     * F10.4,6X,*THE AVERAGE DISTANCE TO THE TURN POINT IS*,F7.2,* NM*)   QUANTA  1372
```

```
  620 FORMAT (* AFTER *,F10.6,* MINUTES, WHEN WEAPON *,I4,* ARRIVES, THE QUANTA   1373
     * RANGES(NM) OF AIRCRAFT ARE */(2X,10F10.4))                        QUANTA   1374
      END                                                                QUANTA   1375
```

```
      FUNCTION CIRINT(R1,R2,D)                                          QUANTA  1376
      DATA PI/3.14159265/                                               QUANTA  1377
C         CIRINT COMPUTES THE INTERSECTION OF TWO CIRCLES WITH RADIUS   QUANTA  1378
C         R1 AND R2.  D IS THE DISTANCE BETWEEN THEM.                   QUANTA  1379
      IF(R1.EQ.0.0 .OR. R2.EQ.0.0) GO TO 1                              QUANTA  1380
      IF(R1+R2.LE.D) GO TO 1                                            QUANTA  1381
      IF(R1+D.LE.R2) GO TO 2                                            QUANTA  1382
      IF(R2+D.LE.R1) GO TO 3                                            QUANTA  1383
C       IF X IS THE DISTANCE BETWEEN THE CENTER OF THE CIRCLE AND THE   QUANTA  1384
C           CORD (THE CORD IS THE LINE SEGEMENT THAT CONNECTS THE       QUANTA  1385
C           POINTS OF INTERSECTION OF THE TWO CIRCLES), THEN THETA      QUANTA  1386
C           (HALF THE ANGLE THAT SWEEPS OUT THE SECTOR), IS THE         QUANTA  1387
C           ARCCOS(X/R) WHERE R IS THE RADIUS.                          QUANTA  1388
C       THEREFORE THE INTERSECTION IS THE SUM OF THE TWO SECTORS        QUANTA  1389
C           (R*R*THETA) MINUS THE RECTANGLE FORMED BY THE SECTORS.      QUANTA  1390
      COST1=0.5*(R1*R1-R2*R2+D*D)/(D*R1)                                QUANTA  1391
      THETA1=ACOS(COST1)                                                QUANTA  1392
      COST2=0.5*(R2*R2-R1*R1+D*D)/(D*R2)                                QUANTA  1393
      THETA2=ACOS(COST2)                                                QUANTA  1394
      CIRINT=R1*R1*THETA1+R2*R2*THETA2-D*R1*SIN(THETA1)                 QUANTA  1395
      RETURN                                                            QUANTA  1396
C         TWO CIRCLES DO NOT INTERSECT                                  QUANTA  1397
    1 CIRINT=0.                                                         QUANTA  1398
      RETURN                                                            QUANTA  1399
C         R2 CIRCLE ENCOMPASSES R1 CIRCLE                               QUANTA  1400
    2 CIRINT=PI*R1*R1                                                   QUANTA  1401
      RETURN                                                            QUANTA  1402
C         R1 CIRCLE ENCOMPASSES R2 CIRCLE                               QUANTA  1403
    3 CIRINT=PI*R2*R2                                                   QUANTA  1404
      RETURN                                                            QUANTA  1405
      END                                                               QUANTA  1406
```

```
      SUBROUTINE SET (I,KSUB)                                           QUANTA  1407
C                                                                       QUANTA  1408
C     SETS BIT TO SIGNIFY PARTICULAR SUB IS OUT OF RANGE OF A PARTICULAR QUANTA  1409
C     BASE.                                                             QUANTA  1410
C     EACH WORD OF BASENO REPRESENTS A BASE                             QUANTA  1411
C     EACH BIT OF BASENO(I) REPRESENTS A SUBS RANGE FROM I-TH BASE      QUANTA  1412
C     EACH BIT (I. E. SUB) IS COUNTED FROM THE LEFT OF WORD BASENO(I)   QUANTA  1413
C                                                                       QUANTA  1414
      COMMON/RNGFLG/BASENO(47),NBITS                                    QUANTA  1415
C                                                                       QUANTA  1416
      TEMP=SHIFT(BASENO(I),KSUB).OR.1B                                  QUANTA  1417
      BASENO(I)=SHIFT(TEMP,NBITS-KSUB)                                  QUANTA  1418
C                                                                       QUANTA  1419
      RETURN                                                            QUANTA  1420
      END                                                               QUANTA  1421
```

```
      SUBROUTINE MARVIN(NMARV,CUT,IMARV)                                QUANTA  1422
      DIMENSION CUT(NMARV)                                              QUANTA  1423
      LEVEL 2, CUT                                                      QUANTA  1424
      DATA WHERE/5HMARVIN/                                              QUANTA  1425
      IMARV=0                                                           QUANTA  1426
      IF(NMARV.LE.0)RETURN                                              QUANTA  1427
      READ(5,5000)(CUT(I),I=1,NMARV)                                    QUANTA  1428
      DO 10 I=1,NMARV                                                   QUANTA  1429
      IF(CUT(I).LT.0.0) CALL ABORT(WHERE,1,"CUT CANNOT BE LESS THAN 0") QUANTA  1430
   10 CONTINUE                                                          QUANTA  1431
      IMARV=1                                                           QUANTA  1432
      WRITE (6,6000) NMARV,NMARV,(CUT(I),I=1,NMARV)                     QUANTA  1433
      CALL PAGE(3)                                                      QUANTA  1434
      RETURN                                                            QUANTA  1435
 5000 FORMAT(7F10.5)                                                    QUANTA  1436
 6000 FORMAT(//14H THERE WILL BE,I6,30H  MARVAL RUN(S) FOR THIS CASE./, QUANTA  1437
     *24H  THE CUT-OFF VALUES ARE,=(2X,F10.5))                          QUANTA  1438
      END                                                               QUANTA  1439
```

```
      SUBROUTINE ALOCIN(NTGTS,NSUBS,NWPNS,MAXNMPS,ALOC,ALOCO,ISUBS,NMPS,
     + ITGTNO)
      DIMENSION ALOC(NTGTS,NWPNS,3),ALOCO(NWPNS),ISUBS(NSUBS),
     + NMPS(NSUBS),ITGTNO(MAXNMPS)
      COMMON /REAL/ ITCUT2,EPSCUT,ITCUT1,CHGKIL
      LEVEL 2,ALOC,ISUBS,NMPS,ITGTNO,ALOCO
      DATA WHERE/6HALOCIN/
      READ(5,5000)CHGKIL,ITCUT1,ITCUT2,EPSCUT
      IF(CHGKIL.LE.0.0) CALL ABORT(WHERE,1,"WHAT IS CONSIDERED TO BE AN
     1INCREASE IN KILL MUST BE GREATER THAN 0")
      IF(EPSCUT.LE.0.0) CALL ABORT(WHERE,2,"THE CONVERGENCE TEST FOR THE
     1 LAGRANGIAN MATRIX MUST BE GREATER THAN 0")
      IF(ITCUT1.LE.0.0) CALL ABORT(WHERE,3,"THE MAXIMUM NUMBER OF ITCUT2
     1 ITERATIONS MUST BE GREATER THAN 0")
      IF(ITCUT2.LE.0.0) CALL ABORT(WHERE,4,"THE NUMBER OF ITERATIONS REQ
     1UIRED TO CHECK FOR INCREASE MUST BE GREATER THAN 0")
C
      DO 20 J=1,NWPNS
      ALOCO(J)=0.0
      DO 20 I=1,NTGTS
      DO 20 K=1,3
      ALOC(I,J,K)=0.
   20 CONTINUE
C                  READ INPUT INITIAL ALLOCATION
      MWPN=0
      DO 50 J=1,NSUBS
      JLIM=ISUBS(J)
      ILIM=NMPS(J)
      DO 50 JSUB=1,JLIM
      READ(5,5010)(ITGTNO(I),I=1,ILIM)
      DO 40 I=1,ILIM
      MWPN=MWPN+1
      II=ITGTNO(I)
      IF((II.LT.-1).OR.(II.GT.NTGTS)) CALL ABORT(WHERE,5,"THE TARGET NUM
     1BER RECEIVING THIS WEAPON MUST BE BETWEEN -1 AND NTGTS")
      IF(II.EQ.0)GO TO 40
      IF(II.EQ.-1)ALOCO(MWPN)=ALOCO(MWPN)+1.
      IF(II.NE.-1) ALOC(II,MWPN,3) =ALOC(II,MWPN,3) +1.
   40 CONTINUE
      IF(JSUB.LT.JLIM)MWPN=MWPN-ILIM
   50 CONTINUE
      RETURN
 5000 FORMAT(F10.4,2I5,F10.4)
 5010 FORMAT(14I5)
      END
```

```
      SUBROUTINE TOTKIL(NTGTS,NWPNS,NTYPES,ALOC,SRVLN,VAL,RELVAL,FLAMB,
     * PROD,VKILL,MANAPB,NACT,ISEQ,SRVPC,BONUS,BASVAL,MTYPE,REL,
     * NMPS,NSUBS,MTYPES,PSB)
      DIMENSION ALOC(NTGTS,NWPNS,3),SRVLN(NWPNS,MANAPB),RELVAL(NTYPES),
     * SRVPC(NWPNS,MANAPB),VAL(NTGTS,NTYPES),FLAMB(NTGTS,NWPNS,3),
     * PROD(NACT),VKILL(NTGTS),ISEQ(NACT),BONUS(NTGTS),BASVAL(NTGTS),
     * MTYPE(NSUBS),REL(MTYPES),NMPS(NSUBS),PSB(MTYPES)
      LEVEL 2,ALOC,SRVPC,SRVLN,VAL,RELVAL,FLAMB,PROD,VKILL,ISEQ,
     * BONUS,BASVAL,MTYPE,REL,NMPS,PSB
C         TOTKIL CALCULATES THE ARRAYS WHICH DEFINE THE OBJECTIVE
C         FUNCTION (PROD,BONUS,VKILL) AND THE LAGRANGIAN MULTIPLIER
C         MATRIX (FLAMB)
      NAC=0
      DO 100 I=1,NTGTS
C         LOCATE FIRST AND LAST AIRCRAFT ON BASE I
      NAC1=NAC+1
      DO 10 JT=1,NTYPES
   10 NAC=NAC+INT(VAL(I,JT)+.1)
      DO 20 IAC=NAC1,NAC
   20 PROD(IAC)=1.0
      BONUS(I)=1.0
C         READ IN SURVIVABILITIES OF AIRCRAFT ON THIS BASE
      NO=2*(I-1)
      CALL READMS(4,SRVLN(1,1),NWPNS*MANAPB,NO+1)
      CALL READMS(4,SRVPC(1,1),NWPNS*MANAPB,NO+2)
C
      J=0
      DO 30 ISUB=1,NSUBS
      MT=MTYPE(ISUB)
      NMIS=NMPS(ISUB)
      DO 30 K=1,NMIS
      J=J+1
      I1=ALOC(I,J,1)
      I2=ALOC(I,J,2)
      I3=ALOC(I,J,3)
C
      IF(I1.NE.0) BONUS(I)=BONUS(I)*(1.-REL(MT))**I1
C
      ISURV=0
      DO 30 IAC=NAC1,NAC
      ISURV=ISURV+1
      IF(I2.NE.0) PROD(IAC)=PROD(IAC)*(SRVLN(J,ISURV)**I2)
      IF(I3.NE.0) PROD(IAC)=PROD(IAC)*(SRVPC(J,ISURV)**I3)
   30 CONTINUE
C         COMPUTE THE VALUE KILLED ON EACH TARGET (VKILL)
      VKILL(I)=0.
      DO 50 IAC=NAC1,NAC
      JT=ISEQ(IAC)
      VKILL(I)=VKILL(I)+RELVAL(JT)*(1.-PROD(IAC))
   50 CONTINUE
```

```
      VKILL(I)=VKILL(I)+BASVAL(I)*(1.-BONUS(I))                         GUANTA    1535
C           COMPUTE THE MULTIPLIER MATRIX (FLAMB)                       GUANTA    1536
      DO 70 J=1,NWPNS                                                   GUANTA    1537
      SUMC=0.                                                           GUANTA    1538
      SUML=0.                                                           GUANTA    1539
      ISURV=0                                                           GUANTA    1540
      DO 60 IAC=NAC1,NAC                                                GUANTA    1541
      ISURV=ISURV+1                                                     GUANTA    1542
      JT=ISEQ(IAC)                                                      GUANTA    1543
      FLAMBI=-PROD(IAC)*ALOG(SRVPC(J,ISURV))*RELVAL(JT)                 GUANTA    1544
      SUMC=SUMC+FLAMBI                                                  GUANTA    1545
      FLAMBI=-PROD(IAC)*ALOG(SRVLN(J,ISURV))*RELVAL(JT)                 GUANTA    1546
      SUML=SUML+FLAMBI                                                  GUANTA    1547
   60 CONTINUE                                                          GUANTA    1548
      FLAMB(I,J,2)=SUML                                                 GUANTA    1549
   70 FLAMB(I,J,3)=SUMC                                                 GUANTA    1550
C           DEFINE THE ARRAY PSB (LN OF THE SURVIV. OF THE BASE)        GUANTA    1551
      DO 80 MT=1,MTYPES                                                 GUANTA    1552
      PSB(MT)=ALOG(1.-REL(MT))                                          GUANTA    1553
   80 CONTINUE                                                          GUANTA    1554
C           COMPUTE ADDITIONAL LAMBDA FOR BASE ATTACK                   GUANTA    1555
      J=0                                                               GUANTA    1556
      DO 90 ISUB=1,NSUBS                                                GUANTA    1557
      MT=MTYPE(ISUB)                                                    GUANTA    1558
      NMIS=NMPS(ISUB)                                                   GUANTA    1559
      DO 90 K=1,NMIS                                                    GUANTA    1560
      J=J+1                                                             GUANTA    1561
   90 FLAMB(I,J)=-(BASVAL(I)*BONUS(I)*PSB(MT))                          GUANTA    1562
C           CONTINUE TO NEXT BASE                                       GUANTA    1563
  100  CONTINUE                                                         GUANTA    1564
      RETURN                                                            GUANTA    1565
      END                                                               GUANTA    1566
```

```
      SUBROUTINE KILOUT(IMARV,OUT,NMARV,NTYPES,SUMV,SUMVK,SUMER,
     * NTGTS,ALOC,VAL,ISEQ,PROD,NACT,RELVAL,VKILL,OBJSUM,ALOCD,NAPNS,
     * IFLAG,BASVAL,BONUS)
      DIMENSION OUT(NMARV),SUMV(NTYPES),SUMVK(NTYPES),SUMER(NTYPES),
     * ALOC(NTGTS,NAPNS,3),VAL(NTGT,NTYPES),ISEQ(NACT),PROD(NACT),
     * RELVAL(NTYPES),VKILL(NTGTS),ALOCD(NAPNS),BASVAL(NTGTS),
     * BONUS(NTGTS)
      LEVEL 2, OUT,SUMV,SUMVK,SUMER,ALOC,VAL,ISEQ,PROD,RELVAL,
     * VKILL,ALOCD,BASVAL,BONUS
C        PRINT HEADER
      CALL PAGE (0)
      CALL PAGE (6)
      IF(NMARV.LE.0) WRITE (6,6000)
      IF(NMARV.GT.0.AND.IFLAG.EQ.0) WRITE(6,6010) IMARV,OUT(IMARV)
      WRITE(6,6020)
C
      DO 110 JT=1,NTYPES
      SUMV(JT)=0.
      SUMVK(JT)=0.
  110 SUMER(JT)=0.
C
      NAC=0
      WAL=0.0
      OBJSUM=0.0
      DO 150 I=1,NTGTS
C        FIND TOTAL WEAPONS SENT TO THREE ATTACK POINTS
      ALBR=0.
      ALLN=0.
      ALPC=0.
      DO 120 J=1,NAPNS
      ALBR=ALBR+ALOC(I,J,1)
      ALLN=ALLN+ALOC(I,J,2)
      ALPC=ALPC+ALOC(I,J,3)
  120 CONTINUE
      BK=1.-BONUS(I)
C        PRINT OUT THOSE TOTALS AND BASE KILL
      WRITE(6,6030) I,ALBR,ALLN,ALPC,BASVAL(I),BK
      CALL PAGE(2)
      WAL=WAL+ALBR+ALLN+ALPC
C
      NACI=NAC+1
      DO 130 JT=1,NTYPES
  130 NAC=NAC+INT(VAL(I,JT)+.1)
      DO 140 IAC=NACI,NAC
      JTYP=ISEQ(IAC)
      VALUE=RELVAL(JTYP)
      EVK=1.-PROD(IAC)
      VK=EVK*VALUE
C        PRINT THE KILL AND IT'S RELATIVE VALUE
      WRITE(6,6040) JTYP,VALUE,VK,EVK
```

```
      CALL PAGE(1)
      SUMV(JITM)=SUMV(JITM)+VALUE
      SUMW(JITM)=SUMW(JITM)+W
      SUM[illegible](JITM)=SUM[illegible](JITM)+[illegible]
  140 CONTINUE
C          COMPUTE AND PRINT TOTAL VALUE OF BASE FILE
      [illegible]
      WRITE(6,5030) [illegible]
      CALL PAGE(1)
  150 CONTINUE
C          PRINT OUT TOTALS
      WRITE (6,5050) [illegible]
      WRITE(6,5060) [illegible]
      WRITE(6,5070) [illegible]
      CALL PAGE [illegible]
C
      IF [illegible] IMARV=IMARV+1
      IF (IMARV.[illegible]) RETURN
      [illegible]=0
      [illegible]=0.
      DO 170 I=1,[illegible]
  170 [illegible]
      [illegible]
      IF (IMARV.[illegible]
      WRITE(6,[illegible])
      CALL PAGE(3)
      RETURN
C
 5000 FORMAT(10X,*THERE ARE NO MARVAL RUNS FOR THIS CASE*/)
 5010 FORMAT(10X,*THIS IS THE NUMBER [illegible] MARVAL RUNS [illegible]*/)
 5020 FORMAT(5X,[illegible]TARGET[illegible],SUM [illegible] OF WEAPONS ALLOCATED [illegible]BASE*
     * [illegible]
     * [illegible]
     * [illegible] FILE*
 5030 FORMAT(/[illegible])
 5040 FORMAT(7X,I3,3F15.4)
 5050 FORMAT(30X,*TARGET*,I4,* HAS A TOTAL KILL OF *,F12.4)
 5060 FORMAT(5X,[illegible]10H TOTALS [illegible])
 5070 FORMAT(/,[illegible] VALUE KILLED [illegible])
 5080 FORMAT(//,* NUMBER OF WEAPONS NOT ALLOCATED [illegible])
      END
```

```
      SUBROUTINE REALOC(NTGTS,NWPNS,NSUBS,NTYPES,NMARV,IMARV,IOUT,
     * SURVHI,VAL,RELVAL,ISUBS,NMPS,ALOC,ALOCD,CUT,FLAMB,VKILL,PROD,
     * OBJSUM,ISEQ,MAXAPB,NACT,BASVAL,BONUS,MTYPE,REL,MTYPES,SURVLO,
     * ASHI,ASLO,XLHI,XLLO,PSB)
C        SSW1 ON TERMINATES RUN .
C        SSW2 ON WRITE RESTART TO DISK .
C        SSW2 OFF WRITE RESTART TAPE .
C
C        TAPE13 IS RESTART TAPE
      DIMENSION SURVHI(NWPNS,MAXAPB),SURVLO(NWPNS,MAXAPB),RELVAL(NTYPES)
     *,ISUBS(NSUBS),NMPS(NSUBS),ALOC(NTGTS,NWPNS,3),FLAMB(NTGTS,NWPNS,3)
     *,CUT(NMARV),ISEQ(NACT),PROD(NACT),VKILL(NTGTS),ALOCD(NWPNS)
     *,BONUS(NTGTS),BASVAL(NTGTS),MTYPE(NSUBS),REL(MTYPES),PSB(MTYPES)
      COMMON /REAL/ ITCUT2,EPSCUT,ITCUT1,CHGKIL
      COMMON/BASTAT/ MNAC,MNACI,LNAC,LNACI
      COMMON/DIFF/ LOW,MAX,JCOL,MAXFL,LOWFL
      COMMON/REC/EPS,J,ISALVO,MSUB,ISSW2,OBHOLD
      COMMON/TRM/ISSW1,IQUIT
      LEVEL 2,SURVHI,VAL,RELVAL,ISUBS,NMPS,ALOC,FLAMB,VKILL,ISEQ,SURVLO,
     * PROD,ALOCD,CUT,BASVAL,BONUS,MTYPE,REL,ASHI,ASLO,XLHI,XLLO,PSB
      DATA J,ISALVO,MSUB/0,0,1/
C           SET UP THE LIMITS ON THE NUMBER OF ITERATIONS
      OBHOLD=0.
      EPS = 0.1
      ITER = 0
      IQUIT = 0
      CALL PAGE(0)
      CALL REMARK(25HSENSE SWITCH1 TERMINATES.)
      CALL SSWTCH(2,ISSW2)
      DO 150 ITER1=1,ITCUT1
      CALL BOSS(0,IMARV,IQUIT)
      DO 130 ITER2=1,ITCUT2
      CALL CONSOLE(25HSENSE SWITCH1 TERMINATES.)
      ITER=ITER+1
      BIC = 0.0
    1 DO 20 JC = 1,NWPNS
C           BEGIN THE SEARCH THROUGH THE NEXT WEAPON GROUP
      J=J+1
      ISALVO=ISALVO+1
C           CHECK FOR THE LAST SALVO (MISSILE) AT THIS SUB LOCATION.
      IF(ISALVO.LE.NMPS(MSUB))GO TO 10
      ISALVO=1
      MSUB=MSUB+1
C           CHECK FOR LAST SUB LOCATION
      IF(MSUB.LE.NSUBS)GO TO 10
      MSUB=1
      J=1
   10 JCOL=J
C           LOOK FOR A MAX AND MIN LAMBDA FROM WEAPON GROUP J
C           SUCH THAT THEIR DIFFERENCE IS GREATER THAN EPS
```

```
      CALL SEARCH(FLAMB,ALOC,NTGTS,NWPNS,RMIN,RMAX,NMARV,CUT,IMARV,    GUANTA  1708
     * ALOCO,MSUB)                                                     GUANTA  1709
      DIF=RMAX-RMIN                                                    GUANTA  1710
      IF(DIF.LT.EPS) GO TO 30                                          GUANTA  1711
      BIG=AMAX1(DIF,BIG)                                               GUANTA  1712
   20 CONTINUE                                                         GUANTA  1713
      WRITE(6,610) ITER,EPS,BIG                                        GUANTA  1714
      CALL PAGE(3)                                                     GUANTA  1715
      EPS = EPS*0.1                                                    GUANTA  1716
      IF (EPS .GE. EPSCUT ) GO TO 1                                    GUANTA  1717
C          AN OPTIMUM HAS BEEN REACHED USING A TOLERANCE OF EPSCUT     GUANTA  1718
      WRITE (6,620) ITER,EPSCUT                                        GUANTA  1719
      CALL PAGE(3)                                                     GUANTA  1720
      GO TO 160                                                        GUANTA  1721
C          A MAXIMUM AND MINIMUM LAMBDA HAS BEEN FOUND.                GUANTA  1722
C          BEGIN THE RE-ALLOCATION PROCESS                             GUANTA  1723
C                                                                      GUANTA  1724
C          CALCULATE LAMBDA FOR EACH INDIVIDUAL AIRCRAFT ON BASE MAX   GUANTA  1725
C          AND BASE LOW.  THIS IS NEEDED TO DEFINE FUNCTION USED IN    GUANTA  1726
C          NEWTON'S ITERATION.                                         GUANTA  1727
   30 CALL GETLAM(VAL,NTGTS,NTYPES,SURVHI,SURVLO,NWPNS,MANAPB,ALHI,ALLO, GUANTA  1728
     * PROD,RELVAL,ISEQ,NACT,ASHI,ASLO)                                GUANTA  1729
C          CALCULATE SURVIVAL AND LAMBDA FOR THE BASES                 GUANTA  1730
      MT=MINO(MSUB)                                                    GUANTA  1731
      BAS=1.-REL(MT)                                                   GUANTA  1732
      ABAS=PSO(MT)                                                     GUANTA  1733
      BONHI=FLAMB(MAX,JCOL)                                            GUANTA  1734
      BONLO=FLAMB(LOW,JCOL)                                            GUANTA  1735
C          ALOW IS THE MAXIMUM NUMBER OF WEAPONS THAT CAN BE TAKEN     GUANTA  1736
C          FROM BASE LOW TO BE PUT ON BASE MAX                         GUANTA  1737
      ALOW=ALOCO(JCOL)                                                 GUANTA  1738
      IF(LOW.NE.-1) ALOW=ALOC(LOW,JCOL,LOWFL)                          GUANTA  1739
      CUTOFF=CUT(IMARV)                                                GUANTA  1740
C          DELTA IS THE NUMBER OF WEAPONS TO BE MOVED FROM LOW TO MAX  GUANTA  1741
      DELTA=XNEWT(0.,ALOW,EPSCUT,SURVHI,SURVLO,ALHI,ALLO,ASHI,ASLO,    GUANTA  1742
     * MANAPB,NWPNS,CUTOFF,BAS,BONHI,BONLO,ABAS)                       GUANTA  1743
C                                                                      GUANTA  1744
      IF(MAX.EQ.-1)GO TO 50                                            GUANTA  1745
      ALOC(MAX,JCOL,MAXFL)=ALOC(MAX,JCOL,MAXFL)+DELTA                  GUANTA  1746
C          CHANGE THE OBJECTIVE FUNCTION AND THE VALUE OF AIRCRAFT     GUANTA  1747
C          KILLED AS A RESULT OF THE NEW ALLOCATION                    GUANTA  1748
      IF(MAXFL.EQ.1) GO TO 40                                          GUANTA  1749
      VR=VRADJ(MNAC1,MNAC,ISEQ,NACT,NTYPES,SURVHI,NWPNS,MANAPB,DELTA,  GUANTA  1750
     * PROD,RELVAL,JCOL)                                               GUANTA  1751
      OBJSUM=OBJSUM+VR-VKILL(MAX)                                      GUANTA  1752
      VKILL(MAX) = VR                                                  GUANTA  1753
C          IF A WEAPON IS BEING MOVED FROM ONE POINT TO ANOTHER POINT  GUANTA  1754
C           ON THE SAME BASE, LAMBDA WILL BE REDEFINED LATER           GUANTA  1755
      IF (MAX.EQ.LOW) GO TO 70                                         GUANTA  1756
C          THE MULTIPLIER MATRIX MUST ALSO BE CHANGED.                 GUANTA  1757
```

```
      CALL ADJLAM(MNACI,MNAC,NWPNS,NTYPES,MXNAPB,ISEQ,PROD,NACT,SURVHI,  QUANTA  1758
     * RELVAL,FLAMB,NTGTS,MAXFL,MAX)                                     QUANTA  1759
      GO TO 55                                                           QUANTA  1760
C          ADJUST OBJECTIVE FCN, BASE SURVIVAL, AND LAMBDA FOR           QUANTA  1761
C          ALLOCATION TO BASE                                            QUANTA  1762
   40 BON=BONUS(MAX)*BAS**DELTA                                          QUANTA  1763
      OBJSUM=OBJSUM+BASVAL(MAX)*(BONUS(MAX)-BON)                         QUANTA  1764
      BONUS(MAX)=BON                                                     QUANTA  1765
C                                                                        QUANTA  1766
      NWO=0                                                              QUANTA  1767
      DO 50 ISUB=1,NSUBS                                                 QUANTA  1768
      M=MTYPE(ISUB)                                                      QUANTA  1769
      BL=BASVAL(MAX)*BONUS(MAX)*PSB(M)                                   QUANTA  1770
      NMIS=NMPS(ISUB)                                                    QUANTA  1771
      DO 50 K=1,NMIS                                                     QUANTA  1772
      NWO=NWO+1                                                          QUANTA  1773
      FLAMB(MAX,NWO,1)=-BL                                               QUANTA  1774
   50 CONTINUE                                                           QUANTA  1775
C                                                                        QUANTA  1776
   55 IF(LOW.EQ.-1)GO TO 110                                             QUANTA  1777
      GO TO 70                                                           QUANTA  1778
C          MAX IS THE DUMMY TARGET                                       QUANTA  1779
   60 ALOCD(JCOL)=ALOCD(JCOL)+DELTA                                      QUANTA  1780
      OBJSUM=OBJSUM+DELTA*CL OFF                                         QUANTA  1781
C          MAKE THE CHANGES ON BASE LOW AS ABOVE                         QUANTA  1782
   70 CHG=-DELTA                                                         QUANTA  1783
      ALOC(LOW,JCOL,LOWFL)=ALOC(LOW,JCOL,LOWFL)+CHG                      QUANTA  1784
      IF(LOWFL.EQ.1) GO TO 80                                            QUANTA  1785
C                                                                        QUANTA  1786
      VK=VKADJ(L,ACI,LNAC,ISEQ,NACT,NTYPES,SURVLO,NWPNS,MXNAPB,CHG,      QUANTA  1787
     * PROD,RELVAL,JCOL)                                                 QUANTA  1788
      OBJSUM=OBJSUM+VK-VKILL(LOW)                                        QUANTA  1789
      VKILL(LOW)=VK                                                      QUANTA  1790
C                                                                        QUANTA  1791
C                                                                        QUANTA  1792
      CALL ADJLAM(LNACI,LNAC,NWPNS,NTYPES,MXNAPB,ISEQ,PROD,NACT,SURVLO,  QUANTA  1793
     *RELVAL,FLAMB,NTGTS,LOWFL,LOW)                                      QUANTA  1794
C                                                                        QUANTA  1795
      GO TO 120                                                          QUANTA  1796
   80 BON=BONUS(LOW)*BAS**CHG                                            QUANTA  1797
      OBJSUM=OBJSUM+BASVAL(LOW)*(BONUS(LOW)-BON)                         QUANTA  1798
      BONUS(LOW)=BON                                                     QUANTA  1799
C                                                                        QUANTA  1800
      NWO=0                                                              QUANTA  1801
      DO 90 ISUB=1,NSUBS                                                 QUANTA  1802
      M=MTYPE(ISUB)                                                      QUANTA  1803
      BL=BASVAL(LOW)*BONUS(LOW)*PSB(M)                                   QUANTA  1804
      NMIS=NMPS(ISUB)                                                    QUANTA  1805
      DO 90 K=1,NMIS                                                     QUANTA  1806
      NWO=NWO+1                                                          QUANTA  1807
```

```
      FLAMB(LOW,NW,1)=-BL
   90 CONTINUE
C
      GO TO 120
C         LOW IS THE DUMMY TARGET
  110 ALOCD(JCOL)=ALOCD(JCOL)-DELTA
      OBJSUM=OBJSUM-DELTA*CUTOFF
C         IF THE USER HAS REQUESTED (IOUT.EQ.2) INFORMATION
C         WILL BE PRINTED EACH TIME A WEAPON IS MOVED
  120 IF(IOUT.EQ.1) GO TO 130
      CALL PAGE(-7)
      WRITE(6,630) ITER
      WRITE(6,640) JCOL,LOW,LOWFL,MAX,MAXFL,DELTA
      WRITE(6,650) OBJSUM
  130 CONTINUE
C         REALOC HAS COMPLETED ITCUT2 ITERATIONS
      IF (IOUT.NE.1) GO TO 140
      CALL PAGE (-5)
      WRITE(6,630) ITER
      WRITE(6,650) OBJSUM
  140 DELOBJ=OBJSUM-OBHOLD
      OBHOLD=OBJSUM
C
      CALL DOSSIT(MARV,IQUIT)
      CALL SSWTCH(1,ISSWI)
      IF (ISSWI.EQ.1 .OR. MOD(ITER1,5).EQ.0 .OR. IQUIT.EQ.1)
     * CALL RSTART(2,IMARV,OBJSUM,ALOC,FLAMB,PROD,VRIL,ALOCD,IGTS,
     * NWPNS,NACT)
      IF(IQUIT.EQ.1 .OR. ISSWI.EQ.1) GO TO 160
C         IF THE OBJECTIVE FUNCTION HAS NOT CHANGED BY AT LEAST
C         CHGRIL (USER SPECIFIED) THE ITERATIVE PROCESS WILL STOP.
      IF (DELOBJ.GE.CHGRIL) GO TO 150
      WRITE(6,660) DELOBJ,CHGRIL,ITCUT1
      CALL PAGE (2)
      GO TO 160
  150 CONTINUE
C         REALOC HAS GONE THROUGH THE MAXIMUM NUMBER OF ITERATIONS:
C         ITCUT1*ITCUT2 (USER SPECIFIED)
      WRITE(6,670) EPS
      CALL PAGE (3)
  160 CONTINUE
      RETURN
C
  610 FORMAT(/* ITERATION *,I4/* THE MULTIPLIER MATRIX HAS CONVERGED*,
     * * USING EPSILON = *,F10.4,*  THE MAXIMUM DELTA LAMBDA FOUND:*,
     * F10.6)
  620 FORMAT(/* ITERATION *,I4/* MULTIPLIER MATRIX CONVERGED WITHIN*,
     * * TOLERANCE OF *,F10.6)
  630 FORMAT (//'7H ITERATION NUMBER,I10)
  640 FORMAT(/15H DELTA N IN COLUMN,I4,9H FROM ROW,I4,*(*,I2,*)*,7H TO R
```

```
      *,I4,5X,F8.2,6X,F8.3,6H  CUR F10.3)
  550 FORMAT (/24H EXPECTED VALUE KILLED =F8.3,/)
  560 FORMAT (/21H KILL CHANGED BY ONLY F7.4,11H (LESS THAN F7.4,5H), I.
     *PASS ,I3,9H ITERATIONS  QUIT.)
  570 FORMAT (/42H ITERATION CUTOFF LIMIT HAS BEEN REACHED.
     */  CURRENT EMSOL IS F8.3)
      END
```

```
          IDENT   CONSOLE                                                   QUANTA   1865
*                                                                           QUANTA   1866
          TITLE   WRITES MESSAGE TO OPERATOR=S CONSOLE ONLY.                QUANTA   1867
          COMMENT WRITES MESSAGE TO OPERATOR=S CONSOLE ONLY.                QUANTA   1868
*                                                                           QUANTA   1869
          ENTRY   CONSOLE                                                   QUANTA   1870
*                                                                           QUANTA   1871
 CONSOLE  EQ      *+1S17           ENTRY/EXIT.                              QUANTA   1872
          MESSAGE X1+1                                                      QUANTA   1873
          EQ      CONSOLE                                                   QUANTA   1874
          END                                                               QUANTA   1875
```

```
         IDENT  PFALTEX                                                  QUANTA   1876
*                                                                        QUANTA   1877
         COMMENT PERMANENT FILE ALTER/EXTEND ROUTINE.  18 MAY 1977.      QUANTA   1878
         TITLE   PERMANENT FILE ALTER/EXTEND ROUTINE.  18 MAY 1977.      QUANTA   1879
*                                                                        QUANTA   1880
*        ENABLES ALTER/EXTEND CONTROL OF THE FOLLOWING PERMANENT FILE -  QUANTA   1881
*               (TAPE13,QRESTRT,ID=DYMAJJB)                              QUANTA   1882
*        N O T E   --   THIS ROUTINE REQUIRES //PFMTEXT//.               QUANTA   1883
*                                                                        QUANTA   1884
         ENTRY  ALTER,EXTEND                                             QUANTA   1885
*                                                                        QUANTA   1886
*        SUBROUTINE ALTER                                                QUANTA   1887
ALTER    EQ     *+1S17         ENTRY/EXIT POINT.                         QUANTA   1888
         ALTER  ALPHA                                                    QUANTA   1889
         EQ     ALTER          RETURN.                                   QUANTA   1890
*                                                                        QUANTA   1891
*        SUBROUTINE EXTEND                                               QUANTA   1892
EXTEND   EQ     *+1S17         ENTRY/EXIT POINT.                         QUANTA   1893
         EXTEND ALPHA                                                    QUANTA   1894
         EQ     EXTEND         RETURN.                                   QUANTA   1895
*                                                                        QUANTA   1896
ALPHA    FDB    TAPE13,QRESTRT,ID=DYMAJJB                                QUANTA   1897
*                                                                        QUANTA   1898
         END                                                             QUANTA   1899
```

```
      SUBROUTINE RSTART (ISTART,IMARV,OBJSUM,ALOC,FLAMB,PROD,VKILL,ALOCD
     *,NTGTS,NWPNS,NACT)
      COMMON/RLC/EPS,J,ISALVO,MSUB,ISSW2,OBHOLD
      DIMENSION ALOC(NTGTS,NWPNS,3), FLAMB(NTGTS,NWPNS,3), PROD(NACT),
     * VKILL(NTGTS), ALOCD(NWPNS)
      INTEGER TODAY,NOW
      LEVEL 2, ALOC,FLAMB,PROD,VKILL,ALOCD
C
      REWIND 13
      IF (ISTART.EQ.2) GO TO 1
C                 READ RESTART TAPE13
      READ (13) TODAY,NOW
      IF (EOF(13).NE.0.0) GO TO 2
      READ(13) IMARV,EPS,J,ISALVO,MSUB,OBHOLD,OBJSUM
      READ (13) (((ALOC(K,L,M),K=1,NTGTS),L=1,NWPNS),M=1,3)
      READ (13) (((FLAMB(K,L,M),K=1,NTGTS),L=1,NWPNS),M=1,3)
      READ (13) (PROD(K),K=1,NACT)
      READ (13) (VKILL(K),K=1,NTGTS)
      READ (13) (ALOCD(K),K=1,NWPNS)
      WRITE (6,3) TODAY,NOW
      GO TO 2
C                 WRITE RESTART TAPE13
    1 CONTINUE
      IF (ISSW2.EQ.1) CALL ALTER
      CALL DATE (TODAY)
      CALL TIME (NOW)
      WRITE (13) TODAY,NOW
      WRITE (13) IMARV,EPS,J,ISALVO,MSUB,OBHOLD,OBJSUM
      WRITE (13) (((ALOC(K,L,M),K=1,NTGTS),L=1,NWPNS),M=1,3)
      WRITE (13) (((FLAMB(K,L,M),K=1,NTGTS),L=1,NWPNS),M=1,3)
      WRITE (13) (PROD(K),K=1,NACT)
      WRITE (13) (VKILL(K),K=1,NTGTS)
      WRITE (13) (ALOCD(K),K=1,NWPNS)
      WRITE (6,4) TODAY,NOW
C
      END FILE 13
      REWIND 13
      IF (ISSW2.EQ.1) CALL EXTEND
C
    2 RETURN
C
    3 FORMAT (5X,26H*** RESTART TAPE READ. ***,5X,2A10)
    4 FORMAT (5X,29H*** RESTART TAPE WRITTEN. ***,5X,2A10)
      END
```

```
      SUBROUTINE TERM (IQUIT,ISSW1)                                           QUANTA   1944
      IF (IQUIT.EQ.1) WRITE (6,1)                                             QUANTA   1945
      IF (ISSW1.EQ.1) WRITE (6,2)                                             QUANTA   1946
      IF (IQUIT.EQ.1) CALL REMARK (33H*** QUANTA TERMINATED BY TIME ***)      QUANTA   1947
      IF (ISSW1.EQ.1) CALL REMARK (33H*** QUANTA TERMINATED BY SSW1 ***)      QUANTA   1948
      STOP                                                                    QUANTA   1949
C                                                                             QUANTA   1950
    1 FORMAT (20X,33H*** QUANTA TERMINATED BY TIME ***)                       QUANTA   1951
    2 FORMAT (20X,33H*** QUANTA TERMINATED BY SSW1 ***)                       QUANTA   1952
      END                                                                     QUANTA   1953
```

```
      SUBROUTINE BUSS (MODE,KIND,IQUIT)                                 GUANTA    1954
C                                                                       GUANTA    1955
C     BUSS CHECKS IF SUFFICIENT TIME REMAINS TO COMPLETE ACTIVITY(KIND), GUANTA    1956
C     BASED ON PREVIOUS TIME REQUIREMENTS, BEFORE LETTING THE ACTIVITY  GUANTA    1957
C     COMMENCE.  ROUTINE BY HARRY M. MURPHY,  2 OCTOBER 1973.           GUANTA    1958
C                                                                       GUANTA    1959
      DIMENSION FN(8), ST(8), STT(8)                                    GUANTA    1960
C                                                                       GUANTA    1961
      DATA FN,ST,STT,KZ,TMIN/8*0.0,8*0.0,8*0.0,0,40./                   GUANTA    1962
C                    START.  GET MODE AND KIND.  CHECK BOTH.            GUANTA    1963
    1 M=MODE                                                            GUANTA    1964
      K=KIND                                                            GUANTA    1965
      IF (K.LT.1.OR.K.GT.8) K=8                                         GUANTA    1966
      IF (M.NE.0) GO TO 3                                               GUANTA    1967
C                    MODE = 0.  ENTERING ACTIVITY(KIND).                GUANTA    1968
      TL=TMIN                                                           GUANTA    1969
      IF (FN(K).LT.1.0) GO TO 2                                         GUANTA    1970
      TM=ST(K)/FN(K)                                                    GUANTA    1971
      TL=TM+TM                                                          GUANTA    1972
      IF (FN(K).LT.2.0) GO TO 2                                         GUANTA    1973
      TS=SQRT(STT(K)/FN(K)-TM*TM)*FN(K)/(FN(K)-1.0)                     GUANTA    1974
      TL=TM+TS+TS                                                       GUANTA    1975
C                    CHECK TIME REMAINING.                              GUANTA    1976
    2 CALL XTIME (TCM,TPP,TIO,TR)                                       GUANTA    1977
      IF(TR.LT.(2.0*TL+TMIN)) IQUIT = 1                                 GUANTA    1978
C                    HAVE SUFFICIENT TIME.  START TIMING ACTIVITY.      GUANTA    1979
      CALL SECOND (TZ)                                                  GUANTA    1980
      KZ=K                                                              GUANTA    1981
      GO TO 5                                                           GUANTA    1982
C                    MODE NON-ZERO.  LEAVING ACTIVITY.  COMPUTE TIME.   GUANTA    1983
    3 IF (K.NE.KZ) GO TO 4                                              GUANTA    1984
      CALL SECOND (TX)                                                  GUANTA    1985
      TL=TX-TZ                                                          GUANTA    1986
      FN(K)=FN(K)+1.0                                                   GUANTA    1987
      ST(K)=ST(K)+TL                                                    GUANTA    1988
      STT(K)=STT(K)+TL*TL                                               GUANTA    1989
    4 KZ=0                                                              GUANTA    1990
C                                                                       GUANTA    1991
    5 RETURN                                                            GUANTA    1992
C                                                                       GUANTA    1993
      END                                                               GUANTA    1994
```

```
      SUBROUTINE SEARCH(FLAMB,ALOC,NTGTS,NWPNS,RMIN,RMAX,NMARV,CUT,    GUANTA  1995
     * IMARV,ALOCD,MSUB)                                                GUANTA  1996
      DIMENSION FLAMB(NTGTS,NWPNS,3),ALOC(NTGTS,NWPNS,3),CUT(NMARV),   GUANTA  1997
     * ALOCD(NWPNS)                                                     GUANTA  1998
      LEVEL 2,FLAMB,ALOC,CUT,ALOCD                                      GUANTA  1999
      COMMON/DIFF/ LOW,MAX,J,MAXFL,LOWFL                                GUANTA  2000
      LOGICAL OUTRNG                                                    GUANTA  2001
C          SEARCH FOR MIN AND MAX LAMBDA FROM WEAPON GROUP J            GUANTA  2002
C          MAX DEFINES THE BASE WITH THE LARGEST VALUE WHILE LOW        GUANTA  2003
C           DEFINES THE BASE WITH THE SMALLEST VALUE                    GUANTA  2004
C          MAXFL AND LOWFL ARE THE ATTACK POINTS ON BASE MAX AND LOW,   GUANTA  2005
C            RESPECTIVELY                                               GUANTA  2006
      RMAX=-1.                                                          GUANTA  2007
      RMIN=999.                                                         GUANTA  2008
      MAX=1                                                             GUANTA  2009
      LOW=1                                                             GUANTA  2010
      DO 20 IF=1,3                                                      GUANTA  2011
      DO 20 I=1,NTGTS                                                   GUANTA  2012
      TEST=FLAMB(I,J,IF)                                                GUANTA  2013
      IF(OUTRNG(I,MSUB).OR.TEST.LE.RMAX) GO TO 10                       GUANTA  2014
      RMAX=TEST                                                         GUANTA  2015
      MAX=I                                                             GUANTA  2016
      MAXFL=IF                                                          GUANTA  2017
C          IF THE NO. OF WEAPONS ON BASE IS NOT LARGE ENOUGH,           GUANTA  2018
C           CONTINUE TO THE NEXT BASE                                   GUANTA  2019
   10 IF(ALOC(I,J,IF).LT..001.OR.TEST.GE.RMIN) GO TO 20                 GUANTA  2020
      RMIN=TEST                                                         GUANTA  2021
      LOW=I                                                             GUANTA  2022
      LOWFL=IF                                                          GUANTA  2023
   20 CONTINUE                                                          GUANTA  2024
      IF(NMARV.LE.0)GO TO 40                                            GUANTA  2025
C          SINCE THIS IS A MARVAL RUN, CHECK OUT THE DUMMY BASE         GUANTA  2026
      CUTOFF=CUT(IMARV)                                                 GUANTA  2027
      IF(CUTOFF.LE.RMAX)GO TO 30                                        GUANTA  2028
      RMAX=CUTOFF                                                       GUANTA  2029
      MAX=-1                                                            GUANTA  2030
   30 IF(ALOCD(J).LT.0.001.OR.CUTOFF.GE.RMIN) GO TO 40                  GUANTA  2031
      RMIN=CUTOFF                                                       GUANTA  2032
      LOW=-1                                                            GUANTA  2033
C                                                                       GUANTA  2034
   40 CONTINUE                                                          GUANTA  2035
      RETURN                                                            GUANTA  2036
      END                                                               GUANTA  2037
```

```
      LOGICAL FUNCTION OUTRNG(I,KSUB)                                   QUANTA 2038
C                                                                       QUANTA 2039
C     TESTS SUB (I. E. BIT )TO SEE IF IT IS OUT OF RANGE.               QUANTA 2040
C                                                                       QUANTA 2041
      COMMON/RNGFLG/BASENO(47),NBITS                                    QUANTA 2042
C                                                                       QUANTA 2043
      TEMP=SHIFT(BASENO(I),KSUB).AND.15                                 QUANTA 2044
      OUTRNG=.FALSE.                                                    QUANTA 2045
      IF(TEMP.NE.0) OUTRNG=.TRUE.                                       QUANTA 2046
C                                                                       QUANTA 2047
      RETURN                                                            QUANTA 2048
      END                                                               QUANTA 2049
```

PAGE 82

```
      SUBROUTINE SETLAM(VAL,NTOTS,NTYPES,SURVHI,SURVLO,NWPNS,MANAPB,ALHI
     * ,ALLO,PROD,RELVAL,ISEQ,NACT,ASHI,ASLO)
      DIMENSION VAL(NTOTS,NTYPES),SURVHI(NWPNS,MANAPB),
     * SURVLO(NWPNS,MANAPB),ALHI(1),ALLO(1),ASHI(1),ASLO(1),
     * PROD(NACT),ISEQ(NACT),RELVAL(NTYPES)
      LEVEL 2,VAL,SURVHI,SURVLO,ALHI,ALLO,ASHI,ASLO,PROD,ISEQ,RELVAL
      COMMON/DIFF/ LOW,MAX,JCOL,MAXFL,LOWFL
      COMMON/BASTAT/ MNAC,MNACI,LNAC,LNACI
C
      IF(MAX.EQ.-1 .OR. MAXFL.EQ.1) GO TO 30
C           LOCATE THE FIRST AND LAST AIRCRAFT OF BASE MAX
      MNAC=0
      DO 10 I=1,MAX
      MNACI=MNAC+1
      DO 10 JT=1,NTYPES
   10 MNAC=MNAC+INT(VAL(I,JT)+.1)
C
      IN=2*(MAX-1)+MAXFL-1
      CALL READMS(4,SURVHI(1,1),NWPNS*MANAPB,IN)
C           CALCULATE LAMBDA FOR EACH AIRCRAFT
      ISRV=0
      DO 20 I=MNACI,MNAC
      ISRV=ISRV+1
      JT=ISEQ(I)
      ASHI(ISRV)=ALOG(SURVHI(JCOL,ISRV))
      ALHI(ISRV)=-PROD(I)*RELVAL(JT)*ASHI(ISRV)
   20 CONTINUE
C
   30 IF(LOW.EQ.-1 .OR. LOWFL.EQ.1) GO TO 60
C          LOCATE THE FIRST AND LAST AIRCRAFT ON BASE LOW
      LNAC=0
      DO 40 I=1,LOW
      LNACI=LNAC+1
      DO 40 JT=1,NTYPES
   40 LNAC=LNAC+INT(VAL(I,JT)+.1)
C
      IN=2*(LOW-1)+LOWFL-1
      CALL READMS(4,SURVLO(1,1),NWPNS*MANAPB,IN)
C          CALCULATE LAMBDA FOR EACH AIRCRAFT
      ISRV=0
      DO 50 I=LNACI,LNAC
      ISRV=ISRV+1
      JT=ISEQ(I)
      ASLO(ISRV)=ALOG(SURVLO(JCOL,ISRV))
      ALLO(ISRV) = -PROD(I)*RELVAL(JT)*ASLO(ISRV)
   50 CONTINUE
C
   60 CONTINUE
      RETURN
      END
```

```
      FUNCTION XNEWT(XMIN,XMAX,EPS,SURVHI,SURVLO,XLHI,XLLO,ASHI,ASLO,    QUANTA   2100
     * MANAPB,NWPNS,CUTOFF,RELM,BONHI,BONLO,AREL)                        QUANTA   2101
      LEVEL 2,SURVHI,SURVLO,XLHI,XLLO,ASHI,ASLO                          QUANTA   2102
      DATA XBND/75./                                                     QUANTA   2103
C                                                                        QUANTA   2104
      CALL ROOT(XMAX,FMAX,DIV,SURVHI,SURVLO,XLHI,XLLO,ASHI,ASLO,MANAPB,  QUANTA   2105
     * NWPNS,CUTOFF,RELM,BONHI,BONLO,AREL)                               QUANTA   2106
      XOLD=XMAX                                                          QUANTA   2107
C         IF THE ROOT LIES TO THE RIGHT OF XMAX, RETURN XMAX             QUANTA   2108
      IF(FMAX.LT.0.0)GO TO 50                                            QUANTA   2109
      IST=0                                                              QUANTA   2110
      KOUNT=0                                                            QUANTA   2111
      IF(DIV.EQ.0.0) GO TO 20                                            QUANTA   2112
      DIFF=FMAX/DIV                                                      QUANTA   2113
   10 XNEW=XOLD-DIFF                                                     QUANTA   2114
      KOUNT=KOUNT+1                                                      QUANTA   2115
      IF(KOUNT.GT.50) GO TO 30                                           QUANTA   2116
      IF(ABS(XNEW).GE.XBND) GO TO 20                                     QUANTA   2117
      XOLD=XNEW                                                          QUANTA   2118
      CALL ROOT(XOLD,FMAX,DIV,SURVHI,SURVLO,XLHI,XLLO,ASHI,ASLO,MANAPB,  QUANTA   2119
     * NWPNS,CUTOFF,RELM,BONHI,BONLO,AREL)                               QUANTA   2120
      IF(DIV.EQ.0.0) GO TO 20                                            QUANTA   2121
      DIFF=FMAX/DIV                                                      QUANTA   2122
      IF(ABS(DIFF).GT.EPS) GO TO 10                                      QUANTA   2123
      XNEWT=XOLD-DIFF                                                    QUANTA   2124
      RETURN                                                             QUANTA   2125
C                                                                        QUANTA   2126
   20 IST=IST+1                                                          QUANTA   2127
      IF(IST.EQ.3) WRITE(6,610)                                          QUANTA   2128
      IF(IST.EQ.3)GO TO 40                                               QUANTA   2129
      XOLD=XMIN                                                          QUANTA   2130
      IF(IST.EQ.2) XOLD=(XMAX+XMIN)/2.                                   QUANTA   2131
      CALL ROOT(XOLD,FMAX,DIV,SURVHI,SURVLO,XLHI,XLLO,ASHI,ASLO,MANAPB,  QUANTA   2132
     * NWPNS,CUTOFF,RELM,BONHI,BONLO,AREL)                               QUANTA   2133
      DIFF=FMAX/DIV                                                      QUANTA   2134
      GO TO 10                                                           QUANTA   2135
C                                                                        QUANTA   2136
   30 WRITE(6,620)                                                       QUANTA   2137
   40 CALL PAGE(1)                                                       QUANTA   2138
      XOLD=(XMAX+XMIN)/2.0                                               QUANTA   2139
   50 XNEWT=XOLD                                                         QUANTA   2140
      RETURN                                                             QUANTA   2141
C                                                                        QUANTA   2142
  610 FORMAT(* NEWTON'S ITERATION DIVERGED 3 TIMES USING XMAX,0.,AND*,   QUANTA   2143
     * * XMID AS INITIAL GUESS........ THE MIDPOINT WAS USED AS ROOT.*)  QUANTA   2144
  620 FORMAT(* NEWTON'S ITERATION FAILED TO CONVERGE AFTER 50 ITERATIONS, QUANTA   2145
     * ........ THE MIDPOINT WAS USED AS ROOT.*)                         QUANTA   2146
      END                                                                QUANTA   2147
```

OE. 64

```
      SUBROUTINE ROOT(X,SUM,DIV,SURVHI,SURVLO,ALHI,ALLO,ASHI,ASLO,MAXARB
     * ,NARNS,CUTOFF,RELM,BUNHI,BUNLO,AREL)
      DIMENSION SURVHI(NARNS,MAXARB),SURVLO(NARNS,MAXARB),ALHI(1),
     * ALLO(1),ASHI(1),ASLO(1)
      LEVEL 2,SURVHI,SURVLO,ALHI,ALLO,ASHI,ASLO
      COMMON/BASTAT/MNAC,MNACI,LNAC,LNACI
      COMMON/DIFF/LOW,MAX,JCOL,MAXFL,LOWFL
      SUM=0.
      PRIME=0.
C        SUM DEFINES THE VALUE OF THE FUNCTION AT X
C        PRIME DEFINES IT'S DERIVATIVE
C
      IF(MAX.EQ.-1)GO TO 30
      IF(MAXFL.EQ.1) GO TO 15
C
      LAST=MNAC-MNACI+1
      DO 10 I=1,LAST
      FCN=ALHI(I)*(SURVHI(JCOL,I)**X)
      SUM=SUM-FCN
      PRIME=PRIME-FCN*ASHI(I)
   10 CONTINUE
      GO TO 20
   15 FCN=BUNHI*(RELM**X)
      SUM=-FCN
      PRIME=-FCN*AREL
C
   20 IF(LOW.EQ.-1)GO TO 50
   30 IF(LOWFL.EQ.1) GO TO 45
C
      LAST=LNAC-LNACI+1
      DO 40 I=1,LAST
      FCN=ALLO(I)*(SURVLO(JCOL,I)**(-X))
      SUM=SUM+FCN
      PRIME=PRIME-FCN*ASLO(I)
   40 CONTINUE
      GO TO 50
   45 FCN=BUNLO*(RELM**(-X))
      SUM=SUM+FCN
      PRIME=PRIME-FCN*AREL
C
   50 IF(MAX.EQ.-1) SUM=SUM-CUTOFF
      IF(LOW.EQ.-1) SUM=CUTOFF+SUM
   60 DIV=PRIME
      RETURN
      END
```

```
      FUNCTION VRADJ(NACI,NAC,ISEG,NACT,NTYPES,SURV,NWPNS,MANARB,DELTA,   GUANTA   2193
     * PROD,RELVAL,JCOL)                                                  GUANTA   2194
      DIMENSION ISEG(NACT),SURV(NWPNS,MANARB),PROD(NACT),RELVAL(NTYPES)   GUANTA   2195
      LEVEL 2,ISEG,SURV,PROD,RELVAL                                       GUANTA   2196
C           VRADJ CALCULATES A NEW PRODUCT (PROD) FOR THOSE AIRCRAFT      GUANTA   2197
C           ON A PARTICULAR BASE, AGAINST WEAPON GROUP JCOL,              GUANTA   2198
C           AND TOTALS THE VALUE KILLED FOR THAT BASE.                    GUANTA   2199
      SUM=0.                                                              GUANTA   2200
      ISR=0                                                               GUANTA   2201
      DO 10 I=NACI,NAC                                                    GUANTA   2202
      ISR=ISR+1                                                           GUANTA   2203
      JT=ISEG(I)                                                          GUANTA   2204
      PROD(I)=PROD(I)*SURV(JCOL,ISR)**DELTA                               GUANTA   2205
      SUM=SUM+RELVAL(JT)*(1.-PROD(I))                                     GUANTA   2206
   10 CONTINUE                                                            GUANTA   2207
      VRADJ=SUM                                                           GUANTA   2208
      RETURN                                                              GUANTA   2209
      END                                                                 GUANTA   2210
```

```
      SUBROUTINE ADJLAM(NAC1,NAC,NWPNS,NTYPES,MXNAPB,ISEQ,PROD,NACT,
     * SURV,RELVAL,FLAMB,NTGTS,IFLAG,IBASE)
      DIMENSION ISEQ(NACT),PROD(NACT),SURV(NWPNS,MXNAPB),RELVAL(NTYPES),
     * FLAMB(NTGTS,NWPNS,3)
      LEVEL 2,ISEQ,PROD,SURV,RELVAL,FLAMB
C           ADJLAM RECALCULATES THE MULTIPLIER MATRIX ACCORDING TO THE
C           CHANGE IN THE PRODUCT MATRIX (WHICH RESULTED FROM THE CHANGE
C           IN ALLOCATION)
      DO 20 J=1,NWPNS
      SUM=0.
      ISR=0
      DO 10 I=NAC1,NAC
      ISR=ISR+1
      JT=ISEQ(I)
      X=-PROD(I)*ALOG(SURV(J,ISR))*RELVAL(JT)
      SUM=SUM+X
   10 CONTINUE
      FLAMB(IBASE,J,IFLAG)=SUM
   20 CONTINUE
C           READ IN THE SURVIVABILITY OF THE AIRCRAFT AGAINST WEAPONS
C           PLACED AT THE OTHER POINT ON BASE.
      IF(IFLAG.EQ.2) GO TO 30
      IN=2*(IBASE-1)+1
      CALL READMS(4,SURV(1,1),NWPNS*MXNAPB,IN)
      I1=2
      GO TO 50
C
   30 IN=2*(IBASE-1)+2
      CALL READMS(4,SURV(1,1),NWPNS*MXNAPB,IN)
      I1=3
C           CALCULATE THE REMAINDER OF THE MULTIPLIER MATRIX
   50 DO 70 J=1,NWPNS
      SUM=0.
      ISR=0
      DO 60 I=NAC1,NAC
      ISR=ISR+1
      JT=ISEQ(I)
      X=-PROD(I)*ALOG(SURV(J,ISR))*RELVAL(JT)
      SUM=SUM+X
   60 CONTINUE
      FLAMB(IBASE,J,I1)=SUM
   70 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ALINT(NTGTS,NWPNS,ALOC,ALOCD)
      DIMENSION ALOC(NTGTS,NWPNS,3),ALOCD(NWPNS)
      LEVEL 2, ALOC,ALOCD
C         INTEGERIZE THE ALLOCATION MATRIX BY WEAPON GROUPS.
C
      DO 60 J=1,NWPNS
C          FIND THE TOTAL NUMBER OF WEAPONS ALLOCATED FROM GROUP J
      ISUM=0
      RSUM=0.
      DO 10 I=1,NTGTS
      ISUM=ISUM+INT(ALOC(I,J,1))+INT(ALOC(I,J,2))+INT(ALOC(I,J,3))
      RSUM=RSUM+ALOC(I,J,1)+ALOC(I,J,2)+ALOC(I,J,3)
   10 CONTINUE
      ISUM = ISUM + INT(ALOCD(J))
      RSUM=RSUM+ALOCD(J)
C          COMPUTE EXCESS NUMBER TO ALLOCATE.
      N=RSUM-FLOAT(ISUM)+.5
      IF(N.EQ.0) GO TO 40
      DO 30 IN=1,N
C          FIND MAXIMUM NUMBER IN ALLOCATION ARRAYS.
      RMAX=0.
      DO 20 IF=1,3
      DO 20 I=1,NTGTS
      TEST=ALOC(I,J,IF)-AINT(ALOC(I,J,IF))
      IF(TEST.LE.RMAX) GO TO 20
      RMAX=TEST
      MAX=I
      LOC=IF
   20 CONTINUE
      TEST = ALOCD(J) - AINT(ALOCD(J))
C          ALLOCATE ONE TO BASE MAX, ATTACK POINT LOC, FROM GROUP J
      IF(TEST.LE.RMAX) ALOC(MAX,J,LOC)=AINT(ALOC(MAX,J,LOC)+1.0)
      IF(TEST.GT.RMAX) ALOCD(J)=AINT(ALOCD(J)+1.0)
   30 CONTINUE
C          FINISH INTEGERIZATION.
   40 DO 50 IF=1,3
      DO 50 I=1,NTGTS
      ALOC(I,J,IF)=AINT(ALOC(I,J,IF))
   50 CONTINUE
      ALOCD(J)=AINT(ALOCD(J))
C          CONTINUE TO THE NEXT WEAPON GROUP.
   60 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ALOUT(NTGTS,NSUBS,NWPNS,ALOC,ISUBS,NMPS)
      DIMENSION ALOC(NTGTS,NWPNS,3),ISUBS(NSUBS),NMPS(NSUBS)
      LEVEL 2, ALOC,ISUBS,NMPS
C         ALOUT PRINTS OUT ALLOCATION INFORMATION.
C
      CALL PAGE (0)
C          PRINT OUT BY TARGET LOCATION.
      WRITE (6,600)
      CALL PAGE (3)
      DO 10 I=1,NTGTS
      CALL PAGE (-2)
      WRITE (6,610) I
      NWG=0
      DO 10 J=1,NSUBS
      NMIS=NMPS(J)
      DO 10 K=1,NMIS
      NWG=NWG+1
      ALBR=ALOC(I,NWG,1)
      ALLN=ALOC(I,NWG,2)
      ALPC=ALOC(I,NWG,3)
      IF(ALBR+ALLN+ALPC.LT..0001) GO TO 10
      WRITE (6,620) J,ISUBS(J),ALBR,ALLN,ALPC,K,NWG
      CALL PAGE (1)
   10 CONTINUE
C          PRINT OUT BY SUBMARINE LOCATION.
      NWG=0
      DO 20 J=1,NSUBS
      CALL PAGE (-2)
      WRITE (6,640) J,ISUBS(J)
      NMIS=NMPS(J)
      DO 20 K=1,NMIS
      NWG=NWG+1
      DO 20 I=1,NTGTS
      ALBR=ALOC(I,NWG,1)
      ALLN=ALOC(I,NWG,2)
      ALPC=ALOC(I,NWG,3)
      A=ALBR+ALLN+ALPC
      IF (A.LT..0001) GO TO 20
      WRITE (6,650) A,K,I,ALBR
      CALL PAGE (1)
   20 CONTINUE
      RETURN
C
  600 FORMAT(6X,25(1H*),*PRINTOUT OF THE OPTIMUM ALLOCATION OF *,
     * *MISSILES*,25(1H*)//)
  610 FORMAT(/6X,*TARGET*,I4,4X,*VS. SUB LOCATION*,10X,*BASE*,4X,
     * *RUNWAY*,3X,*TURN PT*)
  620 FORMAT(20X,I7,* (*,I2,* SUBS)*,4X,3F10.1,* FROM SALVO*,I3,
     * * (WEAPON GROUP*,I4,* )*)
  640 FORMAT(/16X,*SUB LOCATION*,I4,* (*,I2,* SUBS)*)
```

```
2348   650  FORMAT(18X,F10.1,* MISSILES FROM SALVO*,I3,* TO TARGET*,I4,       QUANTA   2349
2349       + * (*,F5.1,* TO BASE)*)                                          QUANTA   2350
2350        END                                                               QUANTA   2351
```

```
      FUNCTION PROFLO (XLO,X,Y,NPTS)                                    QUANTA  2352
      DIMENSION X(NPTS), Y(NPTS)                                        QUANTA  2353
      LEVEL 2,X,Y                                                       QUANTA  2354
      IF (XLO.LE.X(NPTS).AND.XLO.GE.X(1)) GO TO 10                      QUANTA  2355
      IF (XLO.GT.X(NPTS)) GO TO 20                                      QUANTA  2356
      PROFLO=Y(1)                                                       QUANTA  2357
      RETURN                                                            QUANTA  2358
C                                                                       QUANTA  2359
   10 N=NPTS                                                            QUANTA  2360
      PROFLO=AUSC(XLO,X,Y,N)                                            QUANTA  2361
      PROFLO=AMAX1(PROFLO,Y(1))                                         QUANTA  2362
      RETURN                                                            QUANTA  2363
C                                                                       QUANTA  2364
   20 SLOPE=(Y(NPTS)-Y(NPTS-1))/(X(NPTS)-X(NPTS-1))                     QUANTA  2365
      PROFLO=Y(NPTS)+SLOPE*(XLO-X(NPTS))                                QUANTA  2366
      RETURN                                                            QUANTA  2367
      END                                                               QUANTA  2368
```

```
      FUNCTION DIST (XLAT,XLNG,YLAT,YLNG)                             QUANTIA   2369
      LEVEL 2,XLAT,XLNG,YLAT,YLNG                                     QUANTIA   2370
C     THIS FUNCTION COMPUTES THE GREAT CIRCLE DISTANCE (IN N.M.) BETWEEN QUANTIA   2371
C     POINTS X AND Y WITH LATITUDES AND LONGITUDES GIVEN IN TERMS OF  QUANTIA   2372
C     RADIANS (NORTH AND WEST ARE POSITIVE).                          QUANTIA   2373
      DATA PI2/1.570796327/                                           QUANTIA   2374
      A=ABS(XLNG-YLNG)                                                QUANTIA   2375
      B=PI2-XLAT                                                      QUANTIA   2376
      C=PI2-YLAT                                                      QUANTIA   2377
      CBMC=COS(B-C)                                                   QUANTIA   2378
      CBPC=COS(B+C)                                                   QUANTIA   2379
      X=(.5*(CBPC+CBMC)) + (.5*COS(A)*(CBMC-CBPC))                    QUANTIA   2380
      DIST=3442.2*ACOS(X)                                             QUANTIA   2381
      RETURN                                                          QUANTIA   2382
      END                                                             QUANTIA   2383
```

```
      SUBROUTINE PAGE (N)                                              QUANTA   2384
C                                                                      QUANTA   2385
      DATA LINE,NP/61,0/                                               QUANTA   2386
C                                                                      QUANTA   2387
      IF(N.EQ.0 .AND. LINE.GT.0) GO TO 1                               QUANTA   2388
      LINE=LINE+IABS(N)                                                QUANTA   2389
      IF(LINE.LT.60) GO TO 2                                           QUANTA   2390
    1 NP=NP+1                                                          QUANTA   2391
      WRITE(6,3) NP                                                    QUANTA   2392
      LINE=0                                                           QUANTA   2393
      IF(N.LT.0) LINE=-N                                               QUANTA   2394
    2 RETURN                                                           QUANTA   2395
    3 FORMAT(1H1,5H PAGE,I4/)                                          QUANTA   2396
      END                                                              QUANTA   2397
```

```
      FUNCTION SSPF (HM)
C
C     ROUTINE TO COMPUTE SOUND SPEED IN METERS/SECOND AS A FUNCTION OF
C     GEOMETRIC ALTITUDE IN METERS.
C     BASED ON DATA PRESENTED IN //U.S. STANDARD ATMOSPHERE, 1962//.
C
C     ROUTINE BY HARRY M. MURPHY, JR.,  29APR71, CORRECTED 30OCT72 (HMM)
C
      DIMENSION ALT(10), TM(10), DTDZ(10)
C
      DATA ALT/-4996.0,0.0,11019.0,20063.0,32162.0,47350.0,52429.0,61591
     1.0,79994.0,90000.0/
C
      DATA TM/320.65,288.15,216.65,216.65,228.65,270.65,270.65,252.65,18
     10.65,180.65/
C
      DATA DTDZ/-6.5052E-3,-6.4888E-3,0.0,9.9182E-4,2.7653E-3,0.0,-1.997
     14E-3,-3.9124E-3,0.0,0.0/
C
C
   10 Z=HM
      I=1
      IF (Z+5000.0) 50,50,20
   20 IF (90000.0-Z) 60,60,30
C
   30 DO 40 I=1,9
      IF (ALT(I+1)-Z) 40,40,50
   40 CONTINUE
      I=10
C
   50 SSPF=20.046796*SQRT(TM(I)+DTDZ(I)*(Z-ALT(I)))
      RETURN
C
   60 SSPF=269.44
      RETURN
      END
```

```
      FUNCTION AOSC (XV,X,Y,NXY)
C
C     INTERPOLATES A TABLE OF NXY X,Y POINTS AT ENTRY POINT, XV.  USING
C     FIRST-ORDER HERMITE, OR OSCULATORY, INTERPOLATION.
C
C     ALGORITHM  ---  EVALUATE A POLYNOMIAL WHICH PASSES THROUGH A PAIR
C     OF POINTS, X2,Y2, AND X3,Y3,  AND WHICH HAS A GIVEN FIRST DERIVA-
C     TIVE AT EACH POINT.   IN THIS CASE,  THE DERIVATIVE AT POINT 2 IS
C     THE MEAN OF THE SLOPES 1-2 AND 2-3, WHILE THE DERIVATIVE AT POINT
C     3 IS THE MEAN OF THE SLOPES 2-3 AND 3-4.
C
C     ROUTINE BASED ON //HERMITE INTERPOLATION//, CACM ALGORITHM 211.
C     BY G.H.SCHUBERT.   (COMM. ACM, OCT 1963.)
C
C     NOTE  --  THE X-ARRAY MUST MONOTONICALLY INCREASE IN VALUE.
C
C     ROUTINE WRITTEN BY HARRY M. MURPHY.  AFWL/DYM.  4 JAN 1977.
C
      DIMENSION X(NXY), Y(NXY)
      LEVEL 2, X,Y
C
      LOGICAL ERROR
      IF(XV.LT.X(NXY)) GO TO 1
      AOSC=Y(NXY)
      GO TO 3
C                       START.  MAKE XV AND NXY LOCAL VARIABLES.
    1 AOSC=0.0
      X0=XV
      L=1
      M=NXY
C                       CHECK TABLE LENGTH, NXY.
      ERROR=M.LT.2
      IF (ERROR) WRITE (6,4)
      IF (ERROR) GO TO 3
C                       DO BINARY SEARCH OF THE ARRAY, X, FOR ENTRY POINT.
      I=(L+M)/2
    2 IF (X0.GT.X(I)) L=I
      IF (X0.LE.X(I)) M=I
      I=(L+M)/2
      IF (L.LT.I) GO TO 2
C                       WE HAVE THE ENTRY POINT.  COMPUTE X2, X3, Y2, AND
C                       Y3 AS LOCAL VARIABLES.
      X2=X(I)
      X3=X(I+1)
      Y2=Y(I)
      Y3=Y(I+1)
C                       COMPUTE SLOPE VALUES.
      S23=(Y3-Y2)/(X3-X2)
      IF(S23.LE. .01) GO TO 5
      S12=S23
```

```
      IF (I.GT.1) S12=(Y(I)-Y(I-1))/(X(I)-X(I-1))                       QUANTA  2484
      S34=S23                                                           QUANTA  2485
      IF (I.LT.(NXY-1)) S34=(Y(I+2)-Y(I+1))/(X(I+2)-X(I+1))             QUANTA  2486
C                 DEFINE SLOPES AT POINTS 2 AND 3.                      QUANTA  2487
      S2=(S12+S23)/2.0                                                  QUANTA  2488
      S3=(S23+S34)/2.0                                                  QUANTA  2489
C                 PERFORM HERMITE INTERPOLATION AT X0.                  QUANTA  2490
      AOSC=((X3-X0)**2*((X2-X0)*(Y2*2.0/(X2-X3)-S2)+Y2)+(X2-X0)**2*((X3-QUANTA  2491
     1X0)*(Y3*2.0/(X3-X2)-S3)+Y3))/(X3-X2)**2                           QUANTA  2492
C                 AND RETURN.                                           QUANTA  2493
    3 RETURN                                                            QUANTA  2494
    5 AOSC=S23*(X0-X2) + Y2                                             QUANTA  2495
      RETURN                                                            QUANTA  2496
C                                                                       QUANTA  2497
    4 FORMAT (1X/48H FUNCTION AOSC ERROR.  TABLE LENGTH LESS THAN 2./1X) QUANTA  2498
      END                                                               QUANTA  2499
```

```
      SUBROUTINE NUCERR(SUBR,IERROR)                                    NUCMOD   2
C     ****************************************************************  NUCMOD   3
C     ****************************************************************  NUCMOD   4
C     *                                                              *  NUCMOD   5
C     *   S P A R K   A N D   S N A R E   N U C L E A R   M O D U L E S   *  NUCMOD   6
C     *                                                              *  NUCMOD   7
C     ****************************************************************  NUCMOD   8
C     ****************************************************************  NUCMOD   9
C     SUBROUTINE NUCERR IS COMMON TO BOTH THE SPARK AND SNARE NUCLEAR   NUCMOD  10
C     MODULES, AS ARE ATMOS, INT1, AND INT2.                            NUCMOD  11
C                                                                       NUCMOD  12
C     ERROR HAS BEEN DETECTED IN EITHER SPARK OR SNARE-PRINT MESSAGE AND NUCMOD  13
C     TERMINATE EXECUTION.                                              NUCMOD  14
C                                                                       NUCMOD  15
      WRITE(6,6000)IERROR,SUBR                                          NUCMOD  16
      STOP 77                                                           NUCMOD  17
 6000 FORMAT(1H ,"****** ERROR NUMBER ",I2," DETECTED IN THE ",A5," NUCL NUCMOD  18
     *EAR ROUTINE--EXECUTION TERMINATED ******")                        NUCMOD  19
      END                                                               NUCMOD  20
```

```
      SUBROUTINE ATMOS(Z,TM,SIGMA,RHO,THETA,DELTA,CA,AMU,K)            NOCMOD  21
C     CALLING SEQUENCE                                                 NOCMOD  22
C     CALL ATMOS(Z,TM,SIGMA,RHO,THETA,DELTA,CA,AMU,K)                  NOCMOD  23
C     Z      = GEOMETRIC ALTITUDE (FT)                                 NOCMOD  24
C     TM     = MOLECULAR SCALE TEMPERATURE (DEGREES RANKINE)           NOCMOD  25
C     SIGMA = RATIO OF DENSITY TO THAT AT SEA LEVEL                    NOCMOD  26
C     RHO    = DENSITY (SLUGS/CU FT)                                   NOCMOD  27
C     THETA = RATIO OF TEMPERATURE TO THAT AT SEA LEVEL                NOCMOD  28
C     DELTA = RATIO OF PRESSURE TO THAT AT SEA LEVEL                   NOCMOD  29
C     CA     = SPEED OF SOUND (FT/SEC)                                 NOCMOD  30
C     AMU    = VISCOSITY COEFFICIENT (LB-SEC/FT**2)                    NOCMOD  31
C     K = 1 NORMAL,                                                    NOCMOD  32
C       = 2 ALTITUDE GREATER THAN 300000. FT.,                         NOCMOD  33
C       = 3 ALTITUDE NEGATIVE,                                         NOCMOD  34
      DIMENSION HPRIMB(11),TMB(11),SIGMAB(11),ALM(11)                  NOCMOD  35
      DATA HPRIMB/0.0,36089.239,65620.997,154199.480,173884.510,       NOCMOD  36
     1259186.350,295275.590,344488.190,524934.380,557742.780,655174.450/NOCMOD  37
     2,TMB/518.688,389.988,389.988,508.788,508.788,298.188,298.188,    NOCMOD  38
     3465.188,2388.188,2568.188,2838.188/,SIGMAB/1.00,2.9706958E-01,   NOCMOD  39
     43.2665751E-02,1.2117870E-03,5.6677311E-04,1.7329155E-05,         NOCMOD  40
     51.7928595E-06,9.3921519E-08,7.7656593E-10,5.6324877E-10,         NOCMOD  41
     62.5726771E-10/,ALM/-0.00356616,0.0,0.0016459,0.0,-0.0024688,0.0, NOCMOD  42
     70.0021945,0.0109728,0.0054864,0.0027432,0.0019202/               NOCMOD  43
      DATA G/0.0187441767/,RE/2.08553E07/,S/198.72/,                   NOCMOD  44
     1AMUZ/3.7372998E-07/,RHOZ/0.0023769/,TMZ/518.688/                 NOCMOD  45
      K=1                                                              NOCMOD  46
      IF (Z) 1,3,2                                                     NOCMOD  47
    1 K=3                                                              NOCMOD  48
      RETURN                                                           NOCMOD  49
    2 IF(Z.GT.300000.)  K=K+1                                          NOCMOD  50
    3 HPRIM=(RE/(RE+Z))*Z                                              NOCMOD  51
      DO 4 M=1,11                                                      NOCMOD  52
      IF(HPRIM-HPRIMB(M)) 5,6,4                                        NOCMOD  53
    4 CONTINUE                                                         NOCMOD  54
      M=12                                                             NOCMOD  55
    5 M=M-1                                                            NOCMOD  56
    6 IF(ALM(M)) 7,8,7                                                 NOCMOD  57
    7 TM=TMB(M)+ALM(M)*(HPRIM-HPRIMB(M))                               NOCMOD  58
      SIGMA=EXP((1.0+(G/ALM(M)))*(ALOG(TMB(M)/TM)))*SIGMAB(M)          NOCMOD  59
      GO TO 9                                                          NOCMOD  60
    8 TM=TMB(M)                                                        NOCMOD  61
      SIGMA=SIGMAB(M)*EXP(-(G*(HPRIM-HPRIMB(M)))/TMB(M))               NOCMOD  62
    9 RHO=RHOZ*SIGMA                                                   NOCMOD  63
      THETA=TM/TMZ                                                     NOCMOD  64
      DELTA=SIGMA*THETA                                                NOCMOD  65
      CA=49.02177*SQRT(TM)                                             NOCMOD  66
      AMU=AMUZ*SQRT(THETA**3)*((TMZ+S)/(TM+S))                         NOCMOD  67
      RETURN                                                           NOCMOD  68
      END                                                              NOCMOD  69
```

```
      SUBROUTINE INT1 (X,NX,XT,YT,R)                                    NUCMOD  70
C     SUBROUTINE INT1 IS A LINEAR INTERPOLATION ROUTINE.                NUCMOD  71
C     GIVEN A VALUE X, INT1 RETURNS THE CORRESPONDING VALUE Y.          NUCMOD  72
C     NX = DIMENSION OF X-TABLE AND Y-TABLE IN CALLING PROGRAM.         NUCMOD  73
C     XT = TABLE OF X-VALUES IN CALLING PROGRAM.                        NUCMOD  74
C     YT = TABLE OF Y-VALUES IN CALLING PROGRAM.                        NUCMOD  75
C     R = RETURN VALUE Y.                                               NUCMOD  76
      DIMENSION XT(NX),YT(NX)                                           NUCMOD  77
C                                                                       NUCMOD  78
      IF (X-XT(1)) 1,8,2                                                NUCMOD  79
    1 R=YT(1)                                                           NUCMOD  80
      CALL NUCERR(6HINT1  ,1)                                           NUCMOD  81
    2 IF (X-XT(NX)) 4,9,3                                               NUCMOD  82
    3 R=YT(NX)                                                          NUCMOD  83
      CALL NUCERR(6HINT1  ,2)                                           NUCMOD  84
    4 DO 5 I=2,NX                                                       NUCMOD  85
      IF (X-XT(I)) 7,6,5                                                NUCMOD  86
    5 CONTINUE                                                          NUCMOD  87
    6 R=YT(I)                                                           NUCMOD  88
      RETURN                                                            NUCMOD  89
    7 R=YT(I-1)+(X-XT(I-1))*(YT(I)-YT(I-1))/(XT(I)-XT(I-1))             NUCMOD  90
      RETURN                                                            NUCMOD  91
    8 R=YT(1)                                                           NUCMOD  92
      RETURN                                                            NUCMOD  93
    9 R=YT(NX)                                                          NUCMOD  94
      RETURN                                                            NUCMOD  95
      END                                                               NUCMOD  96
```

```
      SUBROUTINE INT2(X,Y,NX,NY,XT,YT,F,FF,NF)                  NUCMOD   97
      DIMENSION F(NF),XT(NX),YT(NY)                             NUCMOD   98
      IF (X-XT(1)) 35,5,5                                       NUCMOD   99
    5 DO 10 I=2,NX                                              NUCMOD  100
      IF (X-XT(I)) 15,15,10                                     NUCMOD  101
   10 CONTINUE                                                  NUCMOD  102
      GO TO 35                                                  NUCMOD  103
   15 IX=I-1                                                    NUCMOD  104
      IF (Y-YT(1)) 40,20,20                                     NUCMOD  105
   20 DO 25 I=2,NY                                              NUCMOD  106
      IF (Y-YT(I)) 30,30,25                                     NUCMOD  107
   25 CONTINUE                                                  NUCMOD  108
      GO TO 40                                                  NUCMOD  109
   30 IY=I-1                                                    NUCMOD  110
      I11=(IX-1)*NY+IY                                          NUCMOD  111
      I12=I11+1                                                 NUCMOD  112
      I21=I11+NY                                                NUCMOD  113
      I22=I21+1                                                 NUCMOD  114
      F11=F(I11)                                                NUCMOD  115
      F12=F(I12)                                                NUCMOD  116
      F21=F(I21)                                                NUCMOD  117
      F22=F(I22)                                                NUCMOD  118
      DY=YT(IY+1)-YT(IY)                                        NUCMOD  119
      DY1=(YT(IY)-Y)/DY                                         NUCMOD  120
      DY2=(YT(IY+1)-Y)/DY                                       NUCMOD  121
      F1=DY2*F11-DY1*F12                                        NUCMOD  122
      F2=DY2*F21-DY1*F22                                        NUCMOD  123
      DX=XT(IX+1)-XT(IX)                                        NUCMOD  124
      DX1=(XT(IX)-X)/DX                                         NUCMOD  125
      DX2=(XT(IX+1)-X)/DX                                       NUCMOD  126
      FF=DX2*F1-DX1*F2                                          NUCMOD  127
      RETURN                                                    NUCMOD  128
   35 CALL NUCERR(6HINT2  ,1)                                   NUCMOD  129
   40 CALL NUCERR(6HINT2  ,2)                                   NUCMOD  130
      END                                                       NUCMOD   31
```

```
      FUNCTION EXT(SMB)                                              NSCMOD 132
      DIMENSION SMBTT(6),EXPT(6)                                     NSCMOD 133
      DATA SMBTT/0.0,.25,.5,1.,1.5,500./                             NSCMOD 134
      DATA EXPT/.625,.639,.642,.575,.809,.809/                       NSCMOD 135
      CALL INTI(SMB,6,SMBTT,EXPT,E)                                  NSCMOD 136
      EXT=E                                                          NSCMOD 137
      RETURN                                                         NSCMOD 138
      END                                                            NSCMOD 139
```

```
      SUBROUTINE SWITCH                                              NOCMOD   140
      COMMON/FLICK/WRT,SF(3),SHB,GO,CTP,S(3),SGOTP,SALT,SHBT,SFT     NOCMOD   141
      SAVE=SF(1)                                                     NOCMOD   142
      SF(1)=S(1)                                                     NOCMOD   143
      S(1)=SAVE                                                      NOCMOD   144
      SAVE=SF(2)                                                     NOCMOD   145
      SF(2)=S(2)                                                     NOCMOD   146
      S(2)=SAVE                                                      NOCMOD   147
      SAVE=SF(3)                                                     NOCMOD   148
      SF(3)=S(3)                                                     NOCMOD   149
      S(3)=SAVE                                                      NOCMOD   150
      RETURN                                                         NOCMOD   151
      END                                                            NOCMOD   152
```

```
      SUBROUTINE GETIR(ALT,HB,HARD,X,T,TR,DELP,DELPR)
      COMMON/FLICK/WKT,SF(3),SHB,GO,CTP,S(3),SOOTP,SALT,SHBT,SFT
      SF=SQRT((ALT-HB)**2+X**2)
      POA= SF(3)*14.695
C     GET IMAGE BURST SLANT RANGE TO FREE AIR POINT
      SRR = SQRT((ALT+HB)**2+X**2)
C     FIND MACH SHOCK STRENGTH AT TPP
   10 XG=SQRT(SRR**2-HB**2)
      IF(XG.LT.GO) GO TO 60
      CALL TPINT(-HB,SRR,HI,GI)
      CALL ATMOS(HI,DUM1,DUM2,DUM3,DUM4,S(3),C,DUM6,DUM7)
      S(1)=(WKT/S(3))**(1./3.)
      S(2)=S(1)*1116.4672/C
      CALL SWITCH
      CALL GETTPP(HI, HB,XTP,SHOZ,TPP,DPM,DPI)
      CALL SWITCH
      POTP=S(3)*14.695
C     GET REFLECTED SHOCK OVERPRESSURE
C     GET SHOCK STRENGTH OF REFLECTED SHOCK AT TPP
      DPRT=DPM-DPI
C     GET IMAGE BURST OVERPRESSURE
      DPRBRA=WFPROP(SRR)
      CALL SWITCH
      DPRBR=WFPROP(SRR)
      CALL SWITCH
C     GET SLANT RANGE OF INCIDENT SHOCK
      SRI=SQRT((HB-HI)**2+GI**2)
C     GET OVERPRESSURE IN WAKE OF INCIDENT SHOCK
   20 TR=WFTR(SRI)
      RZP = WFZR(TR)
      PRAD=WFPR(TR,RZP)
      DPPR=WFPROP(PRAD)
      IF(SR.GT.PRAD) PRAD=SR*1.000001
      DPITAR = WFPRMI(TR,SR,PRAD,DPPR,RZP)
      EXCPT=1.0+DPRT/(POTP+DPI)
      EXCR=1.0+DPRBR/POTP
      FAC=EXCPT/EXCR
      DPTA=((DPITAR+POA)*FAC*(1.0+DPRBRA/POA))-POA
      IF(DPTA.LE.HARD) GO TO 50
      IF(ABS(DPTA-HARD).LT.0.01*HARD) GO TO 50
      SRR=SRR*(DPTA/HARD)**.25
      X=SQRT(SRR**2-(HB+ALT)**2)
      SR=SQRT((ALT-HB)**2+X**2)
      GO TO 10
   50 RETURN
   60 S(1)=WKT**(1./3.)
      S(2)=S(1)
      S(3)=1.0
      POTP=S(3)*14.695
      CALL SWITCH
```

```
      DPI=WFPROP(SRR)                                        NOCMOD  203
      CALL SWITCH                                            NOCMOD  204
      SRG=SRR/S(1)*0.001                                     NOCMOD  205
      CALL CRFR(SRG,RFR)                                     NOCMOD  206
      S(1)=S(1)*RFR**1.4                                     NOCMOD  207
      S(2)=S(1)                                              NOCMOD  208
      CALL SWITCH                                            NOCMOD  209
      DPM=WFPROP(SRR)                                        NOCMOD  210
      CALL SWITCH                                            NOCMOD  211
      DPRT=DPM-DPI                                           NOCMOD  212
      DPRBPA=WFPROP(SRR)                                     NOCMOD  213
      DPRRR=DPI                                              NOCMOD  214
      SRI=SRR                                                NOCMOD  215
      GO TO 20                                               NOCMOD  216
      END                                                    NOCMOD  217
```

```
      SUBROUTINE CRFR(SR,RFR)                                           NOCMOD 218
      DIMENSION FRRR(6),SRT(6)                                          NOCMOD 219
C                                                                       NOCMOD 220
C     DATA FOR RFR                                                      NOCMOD 221
C                                                                       NOCMOD 222
      DATA FRRR/1.96,1.585,1.515,1.460,1.430,1.402833/                  NOCMOD 223
      DATA SRT/0.0,0.9,1.2,1.6,2.0,2.4/                                 NOCMOD 224
      IF(SR.GE.2.4) GO TO 20                                            NOCMOD 225
      CALL INT1(SR,6,SRT,FRRR,RFR)                                      NOCMOD 226
      GO TO 30                                                          NOCMOD 227
   20 RFR=1.3+0.48/SR                                                   NOCMOD 228
   30 RETURN                                                            NOCMOD 229
      END                                                               NOCMOD 230
```

```
      FUNCTION WFZR (TI)
C
C     CALCULATES THE WAVEFORM RADIUS AT ZERO OVERPRESSURE,
C     (WFZR).  (I.E., THE RADIUS SEPARATING THE NEGATIVE AND
C     POSITIVE PHASE PORTIONS OF THE WAVEFORM, AT THE
C     SPECIFIED TIME (T).
C
C        WFZR IS NOT DEFINED FOR TIMES LESS THAN
C        APPROXIMATELY 0.1 SEC BECAUSE FOR THESE
C        TIMES THE WAVEFORM DOES NOT EXHIBIT
C        A NEGATIVE PHASE
C
C           T - TIME(SEC)
C        WFZR - RADIUS(CM)
C
C     THIS ROUTINE IS PART OF THE AFWL 1-KT STANDARD BY NEEDHAM, ET AL.
C
      COMMON/FLICK/WKT,SF(3),SHB,GO,CTP,S(3),SGOTP,SALT,SHBT,SFT
      DATA B/0.032917/,C/-1.086/,C2/33897.0/,B2/8490.0/
      T = TI/SF(2)
C
    1 WFZR=0.0
      IF (T.LT.0.0) CALL NUCERR(4HWFZR,1)
      IF (T.GT.0.0) WFZR=(1.0-B*T**C)*(C2*T+B2)
      RETURN
      END
```

```
      FUNCTION WFPR (TI,RZP)
C
C     CALCULATES THE WAVEFORM RADIUS AT PEAK OVERPRESSURE (WFPR), I.E.,
C     THE RADIUS OF THE SHOCK FRONT, AT THE SPECIFIED TIME (T).
C
C            T = TIME(SEC)
C         WFPR = RADIUS(FT)
C
C     THIS ROUTINE IS PART OF THE AFWL 1-KT STANDARD BY NEEDHAM, ET AL.
C
C
      COMMON/FLICK/WKT,SF(3),SMB,GO,CTP,S(3),SGOTP,SALT,SMBT,SFT
      DATA TA3/-26.25/,TPWR1/0.371/,TPWR2/0.79/
C
      T = TI/SF(2)
    1 IF(T.LT.0.) CALL NUCERR(4HWFPR,1)
      WFPR=0.
      IF (T.EQ.0.0) GO TO 2
C                  CALCULATE RADIUS EXPLICITLY.
C                  EARLY TIME FORM.
      IF (T.LT.0.28) WFPR=24210.*T**TPWR1*(1.+(1.23*T+0.123)*
     1(1.0-EXP(TA3*T**TPWR2)))
      IF (T.LT.0.1) GO TO 2
C                  LATE TIME FORM.
      ALT=ALOG(T)
      ALFT=ALT+3.39182
      RNEW=RZP+1.045E4+2.15E3*ALT+(9.E3/ALFT-6.8E3)/ALFT
C                  INTERPEDIATE TIME INTERPOLATION.
      IF (T.LT.0.28) RNEW=(RNEW*(T-0.1)+WFPR*(0.28-T))/0.18
      WFPR=RNEW
C
    2 WFPR=WFPR*SF(1)/30.48
      RETURN
      END
```

PAGE 12

```
      FUNCTION WFPRMT(TI,RI,PRI ,UPI,RZP)
C
C     THIS ROUTINE IS PART OF THE AFWL 1-KT STANDARD BY NEEDHAM, ET AL.
C
      COMMON/FLICK/WKT,SF(3),SHE,GO,CTP,S(3),SGOTP,SALT,SHBT,SFT
C
      DATA TTOLD/0.0/
C
      T=TI/SF(2)
      R=RI*30.48/SF(1)
      PRAD=PRI*30.48/SF(1)
      UPPK=UPI/SF(3)/1.45038E-5
    1 RPK=PRAD
      IF (T.LT.0.1) GO TO 3
C
      RPK=RPK*1.E-5
      R=R*1.E-5
      IF (T.EQ.TTOLD) GO TO 2
C
      RZ=RZP
      RMN=RZ-9.7163E3*T**.12115
      RZ=RZ*1.E-5
      RMN=RMN*1.E-5
      UPMN=2.2E4/(T*SQRT(T))-0.5*UPPK
      RNEG=RZ-RMN
      RPLS=RPK-RZ
      RNP=RPK-RMN
      ALN=UPPK/RPLS
      BLN=UPPK-ALN*RPK
      UPMNLN=ALN*RMN+BLN
      FMLT=ABS(UPMN/UPPK)
      UPMHY=UPMN+FMLT*(UPMNLN-UPMN)
      ALPHA=(RNP/(UPMHY-UPPK)+RPLS/UPPK)/RNEG
      BETA=-RPLS*(ALPHA+1.0/UPPK)
      FNGZ=RNP/RPLS
      UNUM=ALPHA*RNP+BETA
      BCRMN=1.0-UPMN/(RNP/UNUM+UPPK)
      CRMNLB=ALOG(BCRMN)
      CGZL=(BETA*(1.0/BCRMN-1.0)/UNUM)/((FNGZ*CRMNLB*RNP**(FNGZ-1.0))*(R
     1NP+UPPK*UNUM))
      CGZ=EXP(CGZL)
      BGZL=CRMNLB/CGZ**(RNP**FNGZ)
      BGZ=EXP(BGZL)
C
    2 RBR=RPK-R
      GR=1.0-BGZ**(CGZ**(RBR**FNGZ))
      IF (R.GT.RZ) GR=(RPK-R)/RPLS*GR+(R-RZ)/RPLS
      HR=RBR/(ALPHA*RBR+BETA)+UPPK
      UPR=GR*HR
      RPK=RPK*1.E5
```

```
      R=R*1.E5                                                          NUCMOD 341
      IF (T.GE.0.95) GO TO 5                                            NUCMOD 342
C                                                                       NUCMOD 343
    3 IF (T.EQ.TTOLD) GO TO 4                                           NUCMOD 344
C                                                                       NUCMOD 345
      A=5.446E4*T**(-1.22)-7.135E5*(1.-T**2)                            NUCMOD 346
      IF (T.LT.0.2) C=7.41E-5*T**(-0.885)                               NUCMOD 347
      IF (T.GE.0.2) C=ALOG(-A/(UPPK-A))/(RZP-RPK)                       NUCMOD 348
      B=(UPPK-A)*EXP(-C*RPK)                                            NUCMOD 349
C                                                                       NUCMOD 350
    4 WFLT=A+B*EXP(C*R)                                                 NUCMOD 351
      IF (T.LT.0.1) UPR=WFLT                                            NUCMOD 352
      IF (T.GE.0.1) UPR=(WFLT*(0.95-T)+UPR*(T-0.1))/.85                 NUCMOD 353
C                                                                       NUCMOD 354
    5 TTOLD=T                                                           NUCMOD 355
      WFPRMT=UPR*1.45038E-5*SF(3)                                       NUCMOD 356
      RETURN                                                            NUCMOD 357
      END                                                               NUCMOD 358
```

```
      SUBROUTINE SPARK(   KCASE,     W,         HB,        ALT,       HG,
     *X,       IABOB,    DELP,      TSA,                              HARD,
     *POINTS,  XFA,      XTPP,      XMS,       TFA,       TTPP,      TMS)
C     SUBROUTINE SPARK AND ITS ASSOCIATED SUBROUTINES HAVE BEEN ADAPTED
C     BY CAPT S.D. BARRETT, AFWL/DYVM, FROM PROGRAM SPARK FURNISHED BY
C     AFWL/DYVS.  LAST PROGRAM SPARK UPDATE IDENT = JULY1977.
C
C                                 PARAMETERS
C
C     THE FOLLOWING INPUT PARAMETERS APPLY TO ALL SPARK CALLS-
C          KCASE =1 INDICATES OVERPRESSURE AND TSA SOLUTION DESIRED
C                =2 INDICATES RANGE AND TSA SOLUTIONS DESIRED
C          W     = THE WEAPON YIELD IN KILOTONS.
C          HB    = THE HEIGHT OF BURST ABOVE THE GROUND IN FEET.
C          ALT   = THE RECEIVER ALTITUDE ABOVE THE GROUND IN FEET.
C          HG    = THE ALTITUDE OF THE GROUND IN FEET ABOVE SEA LEVEL.
C
C     THE FOLLOWING PARAMETERS APPLY ONLY WHEN KCASE=1
C     INPUT-
C          X     = THE HORIZONTAL RANGE IN FEET FROM BURST TO RECEIVER
C     OUTPUT-
C          IABOB =0 INDICATES RECEIVER INTERCEPTED BY FREE AIR SHOCK.
C                =1 INDICATES RECEIVER INTERCEPTED BY MACH SHOCK
C          DELP  = PEAK OVERPRESSURE AT THE RECEIVER IN PSI.
C          TSA   = TIME OF SHOCK ARRIVAL AT THE RECEIVER IN SECONDS.
C     THE FOLLOWING PARAMETERS APPLY ONLY WHEN KCASE=2
C     INPUT-
C          HARD  = THE PEAK OVERPRESSURE IN PSI OF INTEREST AT RECEIVER.
C     OUTPUT-
C          POINTS= A FLAG INDICATING THE RADII OF AREAS IN THE
C                  RECEIVER"S ALTITUDE PLANE WHERE THE PEAK OVERPRESSURE
C                  EXCEEDS HARD
C                =1 INDICATES CIRCLE OF RADIUS XFA.
C                =2 INDICATES CIRCLE OF RADIUS XMS.
C                =3 INDICATES CIRCLE OF RADIUS XFA SURROUNDED BY AN
C                     ANNULUS OF MAJOR RADIUS XMS AND MINOR RADIUS XTPP.
C                =0 INDICATES NO AREA.
C          XFA   = HORIZONTAL RANGE IN FEET TO POINT AT WHICH FREE AIR
C                  PEAK OVERPRESSURE EQUALS HARD.
C          XTPP  = HORIZONTAL RANGE IN FEET TO INTERSECTION OF TRIPLE
C                  POINT PATH
C          XMS   = HORIZONTAL RANGE IN FEET TO POINT AT WHICH MACH
C                  REGION PEAK OVERPRESSURE EQUALS HARD.
C          TFA   = TIME OF FREE AIR SHOCK ARRIVAL AT XFA IN SECONDS.
C          TPP   = TIME OF TRIPLE POINT PATH SHOCK ARRIVAL AT XTPP (SEC)
C          TMS   = TIME OF MACH SHOCK ARRIVAL AT XMS IN SECONDS.
      LOGICAL VALID
C     INITIALIZE BLAST AND ATMOSPHERIC PARAMETERS
      HOB=HB+HG
      KGRD=1
```

```
      ACALT=ALT+HB                                                      NUCMOD   409
      CALL ELINIT(HOB,W,ACALT,HB,KGRD)                                  NUCMOD   410
C                                                                       NUCMOD   411
C     IF KCASE=1 FIND DELP AND TSA AT RECEIVER                          NUCMOD   412
C                                                                       NUCMOD   413
      IF(KCASE.NE.1)GO TO 10                                            NUCMOD   414
      SR=SQRT(X**2+(HB-ALT)**2)                                         NUCMOD   415
      CALL BLAST(X,ALT,SR,HB,KGRD,IABOB,TSA,DELP)                       NUCMOD   416
      RETURN                                                            NUCMOD   417
C                                                                       NUCMOD   418
C     IF KCASE=2 FIND RANGE AND TSA SOLUTIONS AT ALT                    NUCMOD   419
C                                                                       NUCMOD   420
C     GET INTERSECTION WITH TRIPLE POINT PATH. TPP                      NUCMOD   421
   10 IF(KGRD.EQ.0)GO TO 20                                             NUCMOD   422
      CALL GETTPP(ALT,HB,XTPP,SRTPP,TTPP,DPR,DPD)                       NUCMOD   423
      IF(DPR.LT.HARD) GO TO 20                                          NUCMOD   424
      CALL GETMS(ALT,HB,HARD,AMS,SRTPP,TMS)                             NUCMOD   425
      POINTS=2.                                                         NUCMOD   426
      RETURN                                                            NUCMOD   427
C     POINTS=1. OR 0.                                                   NUCMOD   428
   20 CALL GETFA(ALT,HB,HARD,XFA,TFA,VALID)                             NUCMOD   429
      POINTS=1.                                                         NUCMOD   430
      IF(.NOT.VALID)POINTS=0.                                           NUCMOD   431
      IF(KGRD.EQ.0.0) GO TO 25                                          NUCMOD   432
      CALL GETTR(ALT,HB,HARD,XFA,TFA,TR,DPI,DPR)                        NUCMOD   433
   25 RETURN                                                            NUCMOD   434
      END                                                               NUCMOD   435
```

```
[illegible]
C [illegible] ALTITUDE, ALT, SET= [illegible]
C [illegible] PRESSURE RATIO TO THAT AT SEA LEVEL
C [illegible] SPEED OF SOUND IN FEET PER SECOND
[illegible]
```

```
      KORD=0                                                           NUCMOD 486
      SF(1) = 1.26*SF(1)                                               NUCMOD 487
      SF(2) = 1.26*SF(2)                                               NUCMOD 488
C     INITIALIZE TRIPLE POINT PARAMETERS IF NECESSARY                  NUCMOD 489
   30 IF(KORD.EQ.0)RETURN                                              NUCMOD 490
C     GET HORIZONTAL RANGE TO TRIPLE POINT PATH (TPP) ORIGIN (GO)      NUCMOD 491
      IF(SHB.LT.2.4)GO TO 40                                           NUCMOD 492
      SGOTP=SHBT**2.43                                                 NUCMOD 493
      GO=SGOTP*SFT*1000.                                               NUCMOD 494
      GO TO 50                                                         NUCMOD 495
   40 CALL INTI(SHBT,14,SHBTP,CSGOTP,SGOTP)                            NUCMOD 496
      GO =SGOTP*SFT*1000.                                              NUCMOD 497
C     GET SCALED TPP "STEEPNESS" FACTOR(CTP)                           NUCMOD 498
   50 IF(SHBT.LT.1.1) GO TO 60                                         NUCMOD 499
      CTP=0.34                                                         NUCMOD 500
      GO TO 70                                                         NUCMOD 501
   60 CALL INTI(SHBT,16,SHBTA,CCTP,CTP)                                NUCMOD 502
   70 SRGOTP=SQRT(SGOTP**2+SHBT**2)                                    NUCMOD 503
      IF(SRGOTP.GE.2.4) GO TO 80                                       NUCMOD 504
      CALL INTI(SRGOTP,6,SRT,FRRR,CRFRRR)                              NUCMOD 505
      GO TO 90                                                         NUCMOD 506
   80 CRFRRR = 1.3+0.48/SRGOTP                                         NUCMOD 507
   90 R = SRGOTP/C1                                                    NUCMOD 508
      S(1) = CRFRRR*SF(1)                                              NUCMOD 509
      S(2) = CRFRRR*SF(2)                                              NUCMOD 510
      FTP = AFRROR(R)                                                  NUCMOD 511
      RETURN                                                           NUCMOD 512
      END                                                              NUCMOD 513
```

```
      FUNCTION AMW(HB,W)                                                  NUCMOD   514
      DIMENSION HUBI(14),BEFFI(14)                                        NUCMOD   515
      DATA(HUBI(J),J=1,14)/0.0,25000.0,50000.0,62500.0,75000.0,875 0.0,   NUCMOD   516
     1100000.0,112500.0,125000.0,137500.0,150000.0,162500.0,175000.0,     NUCMOD   517
     3250000.0/                                                           NUCMOD   518
      DATA(BEFFI(J),J=1,14)/1.0,1.0,0.98,0.96,0.917,0.83,0.66,0.47,0.343  NUCMOD   519
     1,0.265,0.211,0.175,0.143,0.0/                                       NUCMOD   520
      CALL INT1(HB,14,HUBI,BEFFI,W)                                       NUCMOD   521
      AMW=W*H                                                             NUCMOD   522
      RETURN                                                              NUCMOD   523
      END                                                                 NUCMOD   524
```

```
      SUBROUTINE BLAST(X,Z,SR,HB,KGRD,IABGB,T,OUT)                       NUCMOD   525
      COMMON/FLICK/WKT,SF(3),SHB,GO,CTP,S(3),SOGTP,SALT,SHBT,SFT          NUCMOD   526
C     IF IGNORING GROUND REFLECTANCE GET FREE AIR SHOCK                   NUCMOD   527
      IF(KGRD.EQ.0)GO TO 10                                               NUCMOD   528
C     IF HORIZONTAL RANGE LESS THAN RANGE TO TPP ORIGIN IGNORE            NUCMOD   529
C     REFLECTANCE                                                         NUCMOD   530
      IF(X.LE.GO)GO TO 10                                                 NUCMOD   531
C     GET HEIGHT OF TPP AT HORIZONTAL RANGE                               NUCMOD   532
      HTP=GO*CTP*(X/GO-1.)**1.6                                           NUCMOD   533
C     IF RECEIVER ABOVE TPP HEIGHT IGNORE GROUND REFLECTANCE              NUCMOD   534
      IF(Z.GE.HTP)GO TO 10                                                NUCMOD   535
C                                                                         NUCMOD   536
C     GET MACH SHOCK                                                      NUCMOD   537
C                                                                         NUCMOD   538
      IABGB=1                                                             NUCMOD   539
C     GET SLANT RANGE FROM GROUND ZERO TO RECEIVER                        NUCMOD   540
      SRGZ=SQRT(X**2+Z**2)                                                NUCMOD   541
C     GET OVERPRESSURE                                                    NUCMOD   542
      OUT=TPPROP(SRGZ,GINT,HINT)                                          NUCMOD   543
C     GET SLANT RANGE FROM BURST TO INTERSECTION POINT                    NUCMOD   544
      RTA=SQRT(GINT**2+(HINT-HB)**2)                                      NUCMOD   545
C     GET TIME OF SHOCK ARRIVAL (T)                                       NUCMOD   546
      T=WFTP(RTA)                                                         NUCMOD   547
      RETURN                                                              NUCMOD   548
C                                                                         NUCMOD   549
C     GET FREE AIR SHOCK                                                  NUCMOD   550
C                                                                         NUCMOD   551
   10 IABGB=0                                                             NUCMOD   552
C     GET TIME OF SHOCK ARRIVAL (T)                                       NUCMOD   553
      T=WFTP(SR)                                                          NUCMOD   554
C     GET OVERPRESSURE (OUT)                                              NUCMOD   555
      OUT=WFPROP(SR)                                                      NUCMOD   556
C     FOR GROUND RECEIVER ADD EFFECT OF SIMULTANEOUS REFLECTED SHOCK      NUCMOD   557
      IF(Z.EQ.0.0.AND.HB.NE.0.0)OUT=OUT*2.*(7.*14.696*SF(3)+4.*OUT)/      NUCMOD   558
     + (7.*14.696*SF(3)+OUT)                                              NUCMOD   559
      RETURN                                                              NUCMOD   560
      END                                                                 NUCMOD   561
```

```
      SUBROUTINE GETTPP(ALT,HB,X,SRGZ,T,DPR,DPD)                        NUCMOD   562
      COMMON/FLICK/WKT,SF(3),SHB,GO,CTP,S(3),SGCTP,SALT,SHBT,SFT        NUCMOD   563
C     GET HORIZONTAL RANGE TO INTERSECTION OF TRIPLE POINT PATH         NUCMOD   564
      E=EXT(SHBT)                                                       NUCMOD   565
      X=GO*((ALT/CTP/GO)**E+1.)                                         NUCMOD   566
C     GET TIME OF SHOCK ARRIVAL AT INTERSECTION                         NUCMOD   567
      SR=SQRT((ALT-HB)**2+X**2)                                         NUCMOD   568
      T=WFTP(SR)                                                        NUCMOD   569
C     GET FREE AIR SHOCK AT INTERSECTION                                NUCMOD   570
      DPD=WFPKOP(SR)                                                    NUCMOD   571
C      GET MACH SHOCK AT INTERSECTION                                   NUCMOD   572
      SRGZ=SQRT(ALT**2+X**2)                                            NUCMOD   573
      DPR=TPPKOP(SRGZ,GINT,HINT)                                        NUCMOD   574
      RETURN                                                            NUCMOD   575
      END                                                               NUCMOD   576
```

```
      SUBROUTINE GETRA(ALT,HB,HARD,X,T,VALID)                           NUCMOD  577
      LOGICAL VALID                                                     NUCMOD  578
      Z=ALT-HB                                                          NUCMOD  579
      SR=ABS(Z)+.01                                                     NUCMOD  580
      VALID=.TRUE.                                                      NUCMOD  581
C     ITERATE TO FIND SLANT RANGE TO HARD                               NUCMOD  582
   10 X=SR**2-Z**2                                                      NUCMOD  583
      IF(X.LT.0.)GO TO 20                                               NUCMOD  584
      DELP=APPROP(SR)                                                   NUCMOD  585
      IF(ABS(DELP-HARD).LT..01*HARD)GO TO 30                            NUCMOD  586
      SR=SR*(DELP/HARD)**.25                                            NUCMOD  587
      GO TO 10                                                          NUCMOD  588
C     NO SOLUTION                                                       NUCMOD  589
   20 VALID=.FALSE.                                                     NUCMOD  590
      X=0.0                                                             NUCMOD  591
      T=0.0                                                             NUCMOD  592
      RETURN                                                            NUCMOD  593
C     SOLUTION FOUND                                                    NUCMOD  594
   30 X=SQRT(X)                                                         NUCMOD  595
      T=AFTP(SR)                                                        NUCMOD  596
      RETURN                                                            NUCMOD  597
      END                                                               NUCMOD  598
```

```
      SUBROUTINE GETMS(ALT,HB,HARD,X,SRTPP,T)                          NUCMOD  599
      COMMON/FLICK/RT,SF(3),SHB,GO,CTP,S(3),SGOTP,SALT,SHBT,SF         NUCMOD  600
      SRGZ=SRTPP                                                        NUCMOD  601
   10 DELP=TPPROP(SRGZ,GINT,HINT)                                       NUCMOD  602
      IF(ABS(DELP-HARD).LT..01*HARD)GO TO 30                            NUCMOD  603
      SRGZ=SRGZ*(DELP/HARD)**.25                                        NUCMOD  604
      IF(SRGZ.LT.SRTPP)GO TO 20                                         NUCMOD  605
      GO TO 10                                                          NUCMOD  606
C     NO SOLUTION                                                       NUCMOD  607
   20 CALL NUCERR(6HGETMS ,1)                                           NUCMOD  608
C     SOLUTION FOUND                                                    NUCMOD  609
   30 X=SQRT(SRGZ**2-ALT**2)                                            NUCMOD  610
C     GET TSA                                                           NUCMOD  611
      SR=SQRT(GINT**2+(HB-HINT)**2)                                     NUCMOD  612
      T=AFTP(SR)                                                        NUCMOD  613
      RETURN                                                            NUCMOD  614
      END                                                               NUCMOD  615
```

PAGE 23

```
      FUNCTION TPPROP(SRGZ,GINT,HINT)                                   NUCMOD     616
      COMMON/FLICK/WKT,SF(3),SHE,GO,CTP,S(3),SOUTP,SALT,SHBTT,SFT       NUCMOD     617
      DIMENSION SHBT(6),SRMT(13),FRMR(78)                               NUCMOD     618
C                                                                       NUCMOD     619
C     DATA FOR CRFRMR                                                   NUCMOD     620
C                                                                       NUCMOD     621
      DATA SHBT/0.0,0.25,0.5,1.0,1.5,2.0/                               NUCMOD     622
      DATA SRMT/0.0,1.5,1.75,2.0,2.5,3.0,4.0,6.0,8.0,10.0,15.,50.,500./ NUCMOD     623
      DATA FRMR/13*1.06,1.2,1.136,1.225,1.371,1.367,1.36,1.357,1.243,1.1NUCMOD     624
     197,1.175,1.13,1.13,1.13,1.2,1.497,1.51,1.524,1.526,1.506,1.507,1.3NUCMOD     625
     281,1.263,1.234,1.2,1.2,1.2,1.55,1.772,1.784,1.796,1.82,1.822,1.848NUCMOD     626
     3,1.757,1.619,1.434,1.4,1.4,1.4,1.06,1.137,1.249,1.368,1.534,1.677,NUCMOD     627
     41.844,2.066,1.948,1.736,1.5,1.5,1.5,1.06,1.073,1.111,1.148,1.271,1NUCMOD     628
     5.4,1.964,2.088,2.063,1.972,1.8,1.8,1.87                           NUCMOD     629
C     GET HORIZONTAL RANGE AND HEIGHT OF TPP AT INTERSECTION OF CIRCLE  NUCMOD     630
C     OF RADIUS SRGZ ABOUT GROUND ZERO                                  NUCMOD     631
      HGZ=0.                                                            NUCMOD     632
      CALL TPINT(HGZ,SRGZ,HINT,GINT)                                    NUCMOD     633
C     GET CUBE ROOT OF REFLECTION FACTOR IN MACH REFLECTION REGION      NUCMOD     634
      SH=AMIN1(SHB,2.0)                                                 NUCMOD     635
      SSRGZ=AMIN1(SRGZ*.001/SF(1),500.)                                 NUCMOD     636
      CALL INT2(SH,SSRGZ,6,13,SHBT,SRMT,FRMR,CRFRMR,78)                 NUCMOD     637
C     GET OVERPRESSURE (OUT)                                            NUCMOD     638
      RTA=SRGZ/CRFRMR                                                   NUCMOD     639
      TPPROP=WFPROP(RTA)                                                NUCMOD     640
      END                                                               NUCMOD     641
```

```
      FUNCTION WFTP(RFT)                                                NUCMOD 642
C                                                                       NUCMOD 643
C     R - RADIUS (FEET)                                                 NUCMOD 644
C     WFTP - TIME AT PEAK (SEC)                                         NUCMOD 645
C                                                                       NUCMOD 646
      COMMON/FLICK/WKT,SF(3),SHB,GO,CTP,S(3),SGOTP,SALT,SHBT,SFT        NUCMOD 647
C     CONVERT FROM FEET TO METERS                                       NUCMOD 648
      R=RFT*.3048/SF(1)                                                 NUCMOD 649
      IF(R.LT.57.0)GO TO 10                                             NUCMOD 650
      X=SQRT(R)                                                         NUCMOD 651
      Y=SQRT(X)                                                         NUCMOD 652
      WFTP=.0029551*R-.06348*Y-50.446/X+287.41/(R**(2./3.))-263.5/(X*Y) NUCMOD 653
      GO TO 20                                                          NUCMOD 654
   10 WFTP=R*(1.74557E-4+R*(-2.016E-5+R*(1.3371E-6+R*(-3.6325E-8+R*     NUCMOD 655
     *(4.5854E-10+R*(-2.01433E-12))))))                                 NUCMOD 656
   20 CONTINUE                                                          NUCMOD 657
      WFTP=WFTP*SF(2)                                                   NUCMOD 658
      IF(WFTP.LT.0.)CALL NUCERR(6HWFTP  ,1)                             NUCMOD 659
      RETURN                                                            NUCMOD 660
      END                                                               NUCMOD 661
```

```
      FUNCTION WFPROP(RFT)                                              NUCMOD
C                                                                       NUCMOD
C     CALCULATES THE WAVEFORM PEAK OVERPRESSURE (WFPROP)                NUCMOD
C     AT THE SPECIFIED RADIUS (R), I.E., THE OVERPRESSURE               NUCMOD
C     AT THE SHOCK FRONT HAVING THE SPECIFIED RADIUS                    NUCMOD
C     (R).                                                              NUCMOD
C                                                                       NUCMOD
C     R = RADIUS (FEET)                                                 NUCMOD
C     WFPROP = PEAK OVERPRESSURE (PSI)                                  NUCMOD
C                                                                       NUCMOD
C     THIS ROUTINE IS PART OF THE AFWL 1-KT STANDARD BY NEEDHAM, ET AL. NUCMOD
C                                                                       NUCMOD
      COMMON/FLICK/WKT,SF(3),SHR,GO,CTP,S(3),SOOTP,SALT,SHBT,SPT        NUCMOD
      DATA AC/3.18E18/,AO/1.00E14/,ASTAR/9.0E9/                         NUCMOD
C                                                                       NUCMOD
C     CONVERT R TO CM                                                   NUCMOD
      R=RFT*30.48/SF(1)                                                 NUCMOD
      RR=1.0/R                                                          NUCMOD
      RT10=2.24517E-5*R                                                 NUCMOD
      CF=SQRT(ALOG(RT10+3.0*EXP(-(SQRT(RT10)/3.0))))                    NUCMOD
      WFPROP=((AC*RR+AO)*RR+ASTAR/CF)*RR                                NUCMOD
C     CONVERT WFPROP FROM DYNES/CM**2 TO PSI                            NUCMOD
      WFPROP=WFPROP*1.45038E-5*SF(3)                                    NUCMOD
      RETURN                                                            NUCMOD
      END                                                               NUCMOD
```

```
      SUBROUTINE TPINT(H,R,HI,GI)                                       NUCMOD   687
C     GIVEN H, R, C, GO, SUBROUTINE TPINT RETURNS HI AND GI.            NUCMOD   688
C     H IS THE HEIGHT OF THE CENTER OF THE CIRCLE                       NUCMOD   689
C     WHOSE INTERSECTION WITH THE TRIPLE POINT PATH IS SOUGHT.          NUCMOD   690
C     R IS THE RADIUS OF THE CIRCLE.                                    NUCMOD   691
C     C IS THE CONSTANT IN THE EQUATION FOR THE HEIGHT OF THE TRIPLE    NUCMOD   692
C     POINT PATH, HT, AS A FUNCTION OF GROUND RANGE,G.                  NUCMOD   693
C               HT=C*GO*(G/GO-1.0)**1.6                                 NUCMOD   694
C     GO IS THE GROUND RANGE AT WHICH THE TRIPLE POINT PATH STARTS.     NUCMOD   695
C     HI IS THE HEIGHT OF THE INTERSECTION.                             NUCMOD   696
C     GI IS THE GROUND RANGE OF THE INTERSECTION.                       NUCMOD   697
      COMMON/FLICK/WKT,SF(3),SHB,GO,CTP,S(3),SGOTP,SALT,SHBT,SFT        NUCMOD   698
      D=CTP                                                             NUCMOD   699
      IF(R.GT.GO)GO TO 1                                                NUCMOD   700
      HI=0.0                                                            NUCMOD   701
      GI=GO                                                             NUCMOD   702
      RETURN                                                            NUCMOD   703
    1 T=D*(D*GO+H)                                                      NUCMOD   704
      G=(T+SQRT(T**2-(1.0+D**2)*((D*GO+H)**2-R**2)))/(1.0+D**2)         NUCMOD   705
      E=EXT(SHBT)                                                       NUCMOD   706
      HT=CTP*GO*(G/GO-1.0)**(1.0/E)                                     NUCMOD   707
      RT=SQRT(G**2+(HT-H)**2)                                           NUCMOD   708
      IF(ABS(RT/R-1.0).LE.1.0E-07) GO TO 2                              NUCMOD   709
      D=HT/(G-GO)                                                       NUCMOD   710
      GO TO 1                                                           NUCMOD   711
    2 HI=HT                                                             NUCMOD   712
      GI=G                                                              NUCMOD   713
      RETURN                                                            NUCMOD   714
      END                                                               NUCMOD   715
```

```
      SUBROUTINE SNARE( HR,        ACALT,    ACVEL,    HANGLE,   VANGLE,  NUCMOD   716
     *         BANGLE,   YIELD,    HOB,      TERHT,    ALBEDO,   VISIBL,  NUCMOD   717
     *         VAPOR,    CLDALT,   CLOUDA,   CLOUDT,   QC,       QD,      NUCMOD   718
     *         QR,       CAL,      HOR,      VALID,    NOTNEW,   WRTFLG)  NUCMOD   719
C     SUBROUTINE SNARE AND ITS ASSOCIATED SUBROUTINES HAVE BEEN ADAPTED   NUCMOD   720
C     BY CAPT S.D. GARRETT, AFWL/DYVM, FROM PROGRAM SNARE FURNISHED BY    NUCMOD   721
C     AFWL/DYVS.                                                          NUCMOD   722
C                                                                         NUCMOD   723
C                               INPUT PARAMETERS                          NUCMOD   724
C                                                                         NUCMOD   725
C     HR     = THE HORIZONTAL RANGE IN FEET FROM BURST TO RECEIVER.       NUCMOD   726
C     ACALT = THE RECEIVER ALTITUDE ABOVE THE GROUND IN FEET.             NUCMOD   727
C     ACVEL = THE RECEIVER VELOCITY IN FEET/SEC RELATIVE TO THE BURST.    NUCMOD   728
C     HANGLE= THE HORIZONTAL ANGLE IN DEGREES CLOCKWISE THAT THE          NUCMOD   729
C             RECEIVER PANEL IS ROTATED FROM PERPENDICULAR TO THE BURST.  NUCMOD   730
C     VANGLE= THE VERTICAL ANGLE IN DEGREES THAT THE RECEIVER PANEL IS    NUCMOD   731
C             ROTATED FROM PERPENDICULAR TO THE BURST.  POSITIVE WHEN     NUCMOD   732
C             NORMAL TO PANEL POINTS ABOVE THE BURST.                     NUCMOD   733
C     BANGLE= THE HORIZONTAL ANGLE IN DEGREES FROM THE NOSE OF THE RE-    NUCMOD   734
C             CEIVER CLOCKWISE TO THE BURST.                              NUCMOD   735
C     YIELD =THE WEAPON YIELD IN KILOTONS.                                NUCMOD   736
C     HOB    = THE HEIGHT OF BURST ABOVE THE GROUND IN FEET.              NUCMOD   737
C     TERHT = THE ALTITUDE OF THE GROUND IN FEET ABOVE SEA LEVEL.         NUCMOD   738
C     ALBEDO= GROUND REFLECTANCE (.3 FOR SAC NORMAL DAY)                  NUCMOD   739
C     VISIBL= VISIBILITY IN STATUTE MILES (10. FOR SAC NORMAL DAY)        NUCMOD   740
C     VAPOR = SEA LEVEL WATER VAPOR PRESSURE (MM HG) (5. FOR SAC NORMAL   NUCMOD   741
C             DAY)                                                        NUCMOD   742
C     CLDALT= ALTITUDE OF CLOUD LAYER ABOVE THE GROUND IN FEET.           NUCMOD   743
C     CLOUDA= ALBEDO OF THE CLOUD.                                        NUCMOD   744
C     CLOUDT= TRANSMISSITIVITY OF THE CLOUD.                              NUCMOD   745
C     NOTNEW= A LOGICAL VARIABLE WHICH IF .TRUE. INDICATES THAT THE       NUCMOD   746
C             SOLUTION SLANT RANGE FOUND FOR THE PREVIOUS PROBLEM SHOULD  NUCMOD   747
C             BE USED AS THE INITIAL ESTIMATE FOR THIS RANGE SOLUTION     NUCMOD   748
C             PROBLEM.                                                    NUCMOD   749
C     WRTFLG= A LOGICAL VARIABLE WHICH IF .TRUE. CAUSES A MESSAGE TO BE   NUCMOD   750
C             WRITTEN WHENEVER THE RECEIVER ENTERS THE FIREBALL AND       NUCMOD   751
C             HENCE A CALORY LEVEL OF 10000 CAL/CM2 IS RETURNED.          NUCMOD   752
C                                                                         NUCMOD   753
C                         OUTPUT PARAMETERS                               NUCMOD   754
C                                                                         NUCMOD   755
C     QC     = TOTAL THERMAL RADIATION INCIDENT ON RECEIVER PANEL IN      NUCMOD   756
C             CAL/CM2.                                                    NUCMOD   757
C     QD     = DIRECT THERMAL RADIATION INCIDENT ON RECEIVER PANEL IN     NUCMOD   758
C             CAL/CM2.                                                    NUCMOD   759
C     QR     = REFLECTED THERMAL RADIATION INCIDENT ON RECEIVER PANEL IN  NUCMOD   760
C             CAL/CM2                                                     NUCMOD   761
C     CAL    = IF POSITIVE FLAGS THAT THE HORIZONTAL RANGE TO THE POINT   NUCMOD   762
C             IN THE ACALT PLANE AT WHICH CAL CAL/CM2 OF THERMAL RADIA-   NUCMOD   763
C             TION IS INCIDENT ON THE RECEIVER PANEL IS DESIRED.          NUCMOD   764
C            = IF NEGATIVE, FLAGS THAT THE THERMAL RADIATION INCIDENT ON  NUCMOD   765
```

```
C             THE RECEIVER PANEL AT THE RECEIVER LOCATION SPECIFIED BY
C             HR AND ACALT IS DESIRED.
C     HOR    = THE HORIZONTAL RANGE IN FEET TO THE POINT AT WHICH THERMAL
C             RADIATION INCIDENT ON THE RECEIVER PANEL EQUALS CAL.
C     VALID = A LOGICAL VARIABLE WHICH FLAGS HOR=0.0 WHEN .FALSE.
C
      COMMON /SNCM1/     AA,        AB,        AC,        AG,        ALT,
     *        AXR,       CALT,      CHEFL,     DFB,       FBALT,     GALT,
     *        GG,        HFB,       ICLOUD,    NWL,       Q,         QRR(14),
     *        RFB,       TC,        TRAD,      VF,        XFB,       YFB,
     *        ZFB,       XNA,       YNA,       ZNA
C
      COMMON /SNCM2/     AOACAV(14),    DKM,                SIGAVE(14),
     *  TCOW(14),        TDIF(14),      TOZ(14),            TRWL(14),
     *  TWATW(14),       WCO2,          WIN(8)
C
      COMMON /SNCM3/     DT,        W,         SIGMA,     TMAX
C
      COMMON /SNCM4/     PAVE
C
      LOGICAL SKIP,VALID,NOTNEW,WRTFLG,INIT
C
      DIMENSION GWL(4,14),SIGLN(4),WAVEL(14)
C
      DATA GWL/0.,.0008,.0223,.2315,.0056,.0319,.0644,.1245,.5225,.5704,
     1.6027,.4744,.1030,.1042,.0769,.0391,.0574,.0588,.0429,.0209,.1011,
     2.0929,.0735,.0382,.0331,.0277,.0238,.0132,.0097,.0084,.0073,.0041,
     3.0431,.0366,.0319,.0180,.0163,.0128,.0113,.0064,.0282,.0183,.0160,
     4.0093,.0086,.0043,.0037,.0022,.0276,.0130,.0109,.0065,.0133,.0045,
     5.0035,.0021/
C
      DATA SIGLN/0.,2.237,4.5036,6.81063/
C
      DATA WAVEL/.24,.3,0.52,0.75,0.835,0.975,1.14,1.225,1.395,1.62,1.89
     1,2.175,2.635,3.285/
C
      DATA PI/3.14159265359/
C
      DATA RADEG/0.01745329252/
C
      DATA NWL,NOWL/14,4/
C
      DATA DELTIM,TSTOP/.05,20./
C
      DATA REST/0.0/
C
      DATA WLAST,HBLAST/-1.,-1./
C
      DATA INIT/.FALSE./
C
```

```
      IF(INIT)GO TO 1                                                   NUCMOD 816
      NOTNEW=.FALSE.                                                    NUCMOD 817
      WRTFLG=.FALSE.                                                    NUCMOD 818
      INIT=.TRUE.                                                       NUCMOD 819
    1 CONTINUE                                                          NUCMOD 820
      HOB = HR                                                          NUCMOD 821
      ALT = ACALT+TERHT                                                 NUCMOD 822
      VEL = ACVEL                                                       NUCMOD 823
      W = YIELD                                                         NUCMOD 824
      BALT=HOB+TERHT                                                    NUCMOD 825
      PHLAG=0.                                                          NUCMOD 826
      VALID=.TRUE.                                                      NUCMOD 827
      GALT = TERHT                                                      NUCMOD 828
      AG = ALBEDO                                                       NUCMOD 829
      CALT = CLDALT                                                     NUCMOD 830
      AC = CLOUDA                                                       NUCMOD 831
      TC = CLOUDT                                                       NUCMOD 832
      DIFALT = HOB - ACALT                                              NUCMOD 833
      IWDN = INT(VAPOR/2.)                                              NUCMOD 834
      BHANG = BANGLE * RADEG                                            NUCMOD 835
      ICLOUD = 0                                                        NUCMOD 836
      IF(AC .GT. 0.) ICLOUD = 1                                         NUCMOD 837
C                                                                       NUCMOD 838
C     INITIALIZE THE ATMOSPHERIC ROUTINES                               NUCMOD 839
C                                                                       NUCMOD 840
      CALL AINT(BALT,ALT,WAVEL,VISIBL,IWDN,PAVE,WIN,SIGAVE,AOACAV,      NUCMOD 841
     *          WCO2)                                                   NUCMOD 842
      CALL ATMOS(BALT,TM,SIGMA,RHO,THETA,DUMMY,CA,AMU,K)                NUCMOD 843
      SIGLNA = -ALOG(SIGMA)                                             NUCMOD 844
      DO 10 IWL=1,NWL                                                   NUCMOD 845
      CALL INT1(SIGLNA,NDWL,SIGLN,QWL(1,IWL),GRR(IWL))                  NUCMOD 846
   10 CONTINUE                                                          NUCMOD 847
      CALL TDIFFS(0)                                                    NUCMOD 848
      CALL TCO2(0)                                                      NUCMOD 849
      CALL TWATER(0)                                                    NUCMOD 850
      RFB = 193.0 * YIELD**0.35/ SIGMA**0.2                             NUCMOD 851
      DLIM=RFB/5.                                                       NUCMOD 852
C                                                                       NUCMOD 853
C     MAKE INITIAL ESTIMATE OF LETHAL RADIUS                            NUCMOD 854
C                                                                       NUCMOD 855
      JJJ=0                                                             NUCMOD 856
      IF(CAL.LE.0.0)GO TO 15                                            NUCMOD 857
      IF(((REST*REST-DIFALT*DIFALT).GT.0.0).AND.NOTNEW)GO TO 14         NUCMOD 858
      NOTNEW=.TRUE.                                                     NUCMOD 859
      REST=.0328*SQRT(3.183E+10*W/CAL)+110.*SQRT(RFB)                   NUCMOD 860
      IF((REST.GT.2.*RFB).AND.(REST.GT.ABS(DIFALT)))GO TO 14            NUCMOD 861
      PHLAG=1.                                                          NUCMOD 862
      REST=AMAX1((2.*RFB),(ABS(DIFALT)+.01))                            NUCMOD 863
C                                                                       NUCMOD 864
C     SET ESTIMATED TARGET POSITION RELATIVE TO BURST                   NUCMOD 865
```

```
C
   14 HOR=SQRT(REST*REST-DIFALT*DIFALT)
   15 BVANG=ATAN2(HOR,DIFALT)
      IF(BVANG.LE.0.)BVANG=BVANG+PI
      XFB=HOR*COS(BHANG)
      YFB=HOR*SIN(BHANG)
      ZFB=DIFALT
      PHANG=BHANG+HANGLE*RADEG
      PVANG=BVANG+VANGLE*RADEG
      XNA=SIN(PVANG)*COS(PHANG)
      YNA=SIN(PVANG)*SIN(PHANG)
      ZNA=COS(PVANG)
C
C     INITIALIZE THE THERMAL RADIATION VARIABLES
C
      JJJ=JJJ+1
      IHEATS=0
      SKIP=.TRUE.
      IF((YIELD.EQ.WLAST).AND.(HOB.EQ.HBLAST))GO TO 5
      SKIP=.FALSE.
      DUMMY=POWER(DELTIM,0)
      WLAST=YIELD
      HBLAST=HOB
    5 CONTINUE
      CREFL = 1.0
      QC = 0.0
      QD = 0.0
      QR = 0.0
      DDD=0.
      FBLAST = 0.0
      TIME = -DELTIM / 2.
C
C     INTEGRATE THE THERMAL POWER OVER TIME
C
   20 TIME = TIME + DELTIM
      IF(TIME .GT. TSTOP) GO TO 30
      TRAD = TIME
      FBRIS = 11.0 * SQRT(RFB) * TIME
      D = FBRIS - FBLAST
      IF(D .GT. DLIM) CREFL = 1.
      FBALT = BALT + FBRIS
      IF((FBALT-GALT) .LT. RFB) CREFL = 1.
      XFB = XFB - VEL * DELTIM
      ZFB=FBALT-ALT
      RFB = FBALT - GALT
      TDIST = TRAD * VEL - DDD
      IF(TDIST .GE. DLIM) CREFL = 1.
      IF(CREFL.NE.1.0)GO TO 25
      DDD=TRAD*VEL
      FBLAST=FBRIS
```

```
   25 CALL XHEAT(IHEATS,SKIP,FBFLAG)
      IF(FBFLAG.EQ.1.)GO TO 45
      QQ = 39.0577 * Q
      QRATE = QQ / DELTIM
      QC = QQ + QC
      QV = QQ / VF
      QD = QD + QV * AA
      QR = QR + QV * AB
      IF(TIME .LT. TMAX) GO TO 20
      IF(QRATE .GE. 0.027) GO TO 20
      CALL QGMM(BALT,W,HOR,REST,QC,QD,QR)
C
C     IF LEVEL SOLUTION ONLY DESIRED, RETURN
C
   30 IF(CAL.LT.0.0)RETURN
C
C     SEE IF Q NEAR CAL.  IF NOT, CHANGE TARGET POSITION AND ITERATE
C
      IF(ABS(QC-CAL).LT.0.01*CAL)RETURN
      IF(JJJ.GT.10)GO TO 40
      REST=GUESN(JJJ,CAL,QC,REST)
      IF((REST.GT.RFB).AND.(REST.GT.ABS(DIFALT)))GO TO 14
      IF(PHLAG.EQ.1.)GO TO 35
      PHLAG=1.
      REST=AMAX1(RFB+.01,(ABS(DIFALT)+.01))
      GO TO 14
C
C     ABNORMAL RETURNS
C
   35 VALID=.FALSE.
      HOR=0.0
      RETURN
C     DIVERGENCE ABORT
   40 CALL NUCERR(6HSNAKE ,1)
C     RECEIVER GOT IN THE FIREBALL
   45 QC=10000.
      QD=10000.
      QR=0.0
      IF(WRTFLG)WRITE(6,6000)
 6000 FORMAT(" RECEIVER ENTERED FIREBALL")
      RETURN
      END
```

```
      SUBROUTINE AHEAT(IHEATS,SKIP,FBFLAG)                              NUCMOD 958
C                                                                       NUCMOD 959
      COMMON /SNCM1/    AA,       AE,       AC,       AG,       ALT,    NUCMOD 960
     *         AAR,     CALT,     CREFL,    DFB,      FBALT,    GALT,   NUCMOD 961
     *         GG,      HFB,      ICLOUD,   NWL,      G,        GRR(14),NUCMOD 962
     *         RFB,     TC,       TRAD,     VF,       XFB,      YFB,    NUCMOD 963
     *         ZFB,     XNA,      YNA,      ZNA                         NUCMOD 964
C                                                                       NUCMOD 965
      COMMON /SNCM2/    AOACAV(14),  DKM,                SIGAVE(14),    NUCMOD 966
     *   TCCW(14),      TDIF(14),    TDZ(14),            TRWL(14),      NUCMOD 967
     *   TWATW(14),     WCO2,        WIN(8)                             NUCMOD 968
C                                                                       NUCMOD 969
      LOGICAL SKIP                                                      NUCMOD 970
C                                                                       NUCMOD 971
      DIMENSION GRADWL(14)                                              NUCMOD 972
      DIMENSION HEATS(400)                                              NUCMOD 973
C                                                                       NUCMOD 974
      DATA PI/3.14159265359/                                            NUCMOD 975
C                                                                       NUCMOD 976
      G=0                                                               NUCMOD 977
      IF(CREFL.EQ.1.)CALL VIEWF(VVVF,FBFLAG)                            NUCMOD 978
      IF(FBFLAG.EQ.1.)RETURN                                            NUCMOD 979
    5 VF = VVVF                                                         NUCMOD 980
      AA=VF                                                             NUCMOD 981
      IF(CREFL .EQ. 1.) CALL TRANS(DFB)                                 NUCMOD 982
      IF(SKIP)GO TO 1                                                   NUCMOD 983
      IHEATS=IHEATS+1                                                   NUCMOD 984
    3 GRAD=POWER(TRAD,1)                                                NUCMOD 985
      HEATS(IHEATS)=GRAD                                                NUCMOD 986
      MAXHT=IHEATS                                                      NUCMOD 987
      GO TO 2                                                           NUCMOD 988
    1 IHEATS=IHEATS+1                                                   NUCMOD 989
      IF(IHEATS.GT.MAXHT)GO TO 3                                        NUCMOD 990
      GRAD=HEATS(IHEATS)                                                NUCMOD 991
    2 CONTINUE                                                          NUCMOD 992
      IF (CREFL.EQ.0.) GO TO 12                                         NUCMOD 993
C                                                                       NUCMOD 994
C     CALCULATE EFFECTIVE ALBEDOS AND GEOMETRIES                        NUCMOD 995
C                                                                       NUCMOD 996
      IF (ICLOUD.EQ.1) GO TO 6                                          NUCMOD 997
      AGE=AG                                                            NUCMOD 998
      ACE=0.0                                                           NUCMOD 999
      GO TO 10                                                          NUCMOD 1000
    6 IF (ALT.GT.CALT) GO TO 8                                          NUCMOD 1001
      IF (FBALT.GT.CALT) GO TO 7                                        NUCMOD 1002
      AGE=AG*(1.0+AC)                                                   NUCMOD 1003
      ACE=AC*(1.0+AG)                                                   NUCMOD 1004
      GO TO 10                                                          NUCMOD 1005
    7 AGE=AG*TC                                                         NUCMOD 1006
      ACE=TC                                                            NUCMOD 1007
```

```
      GO TO 10                                                          NUCMOD  1008
    8 IF (FBALT.GT.CALT) GO TO 9                                        NUCMOD  1009
      AGE=0.0                                                           NUCMOD  1010
      ACE=TC*(1.0+AG)                                                   NUCMOD  1011
      GO TO 10                                                          NUCMOD  1012
    9 AGE=0.0                                                           NUCMOD  1013
      ACE=AC                                                            NUCMOD  1014
   10 IF (AGE.EQ.0.0) GO TO 11                                          NUCMOD  1015
      HFBG=HFB                                                          NUCMOD  1016
      IF (HFB.LT.RFB) HFBG=HFB+0.22*(RFB-HFB)                           NUCMOD  1017
      HHG=(ALT-GALT)/HFBG                                               NUCMOD  1018
      GGG=GG/HFBG                                                       NUCMOD  1019
C                                                                       NUCMOD  1020
C     SET UP CORRECTION FACTOR TO ACCOUNT FOR FACT THAT                 NUCMOD  1021
C     REFLECTION RATIO RETURNED BY REFL WILL BE                         NUCMOD  1022
C     NORMALIZED TO DIRECT RADIATION INTENSITY FROM                     NUCMOD  1023
C     BURST AT EQUIVALENT HEIGHT, HFBG, INSTEAD                         NUCMOD  1024
C     OF BURST AT ACTUAL HEIGHT, HFB.                                   NUCMOD  1025
C                                                                       NUCMOD  1026
      RRG=1.0                                                           NUCMOD  1027
      IF (HFB.LT.RFB) RRG=((HHG-HFB/HFBG)**2+GGG**2)/((HHG-1.0)**2+GGG**NUCMOD  1028
     12)                                                                NUCMOD  1029
   11 IF (ACE.EQ.0.0) GO TO 12                                          NUCMOD  1030
      HFBC=ABS(FBALT-CALT)                                              NUCMOD  1031
      IF (HFBC.LT.RFB) HFBC=HFBC+0.22*(RFB-HFBC)                        NUCMOD  1032
      HHC=ABS(ALT-CALT)/HFBC                                            NUCMOD  1033
      GGC=GG/HFBC                                                       NUCMOD  1034
C                                                                       NUCMOD  1035
C     SET UP CORRECTION FACTOR TO ACCOUNT FOR FACT THAT                 NUCMOD  1036
C     REFLECTION RATIO RETURNED BY REFL WILL BE                         NUCMOD  1037
C     NORMALIZED TO DIRECT RADIATION INTENSITY FROM                     NUCMOD  1038
C     BURST AT EQUIVALENT HEIGHT, HFBC, INSTEAD                         NUCMOD  1039
C     OF BURST AT ACTUAL HEIGHT, HFB.                                   NUCMOD  1040
C                                                                       NUCMOD  1041
      RRC=1.0                                                           NUCMOD  1042
      IF (HFBC.LT.RFB) RRC=((HHC-ABS(FBALT-CALT)/HFBC)**2+GGC**2)/((HHC-NUCMOD  1043
     11.0)**2+GGC**2)                                                   NUCMOD  1044
   12 DO 13 IWL=1,NWL                                                   NUCMOD  1045
   13 GRADWL(IWL)=GRAD*GRR(IWL)*TRWL(IWL)                               NUCMOD  1046
      IF (AGE+ACE.EQ.0.0) GO TO 15                                      NUCMOD  1047
      IF (AGE.EQ.0.0) GO TO 14                                          NUCMOD  1048
      IF(CREFL.EQ.1.)CALL REFL(GGG,HHG,RR)                              NUCMOD  1049
      VF=VF+RR*RRG*AGE                                                  NUCMOD  1050
   14 IF (ACE.EQ.0.0) GO TO 15                                          NUCMOD  1051
      IF(CREFL.EQ.1.)CALL REFL(GGC,HHC,RR)                              NUCMOD  1052
      VF=VF+RR*RRC*ACE                                                  NUCMOD  1053
   15 CREFL=0.                                                          NUCMOD  1054
      DO 16 IWL=1,NWL                                                   NUCMOD  1055
   16 Q=Q+GRADWL(IWL)*VF/(576.0*PI*DFB**2)                              NUCMOD  1056
      AB=VF-AA                                                          NUCMOD  1057
```

```
1057        RETURN                                        NUCMOD  1058
1058        END                                           NUCMOD  1059
```

```
      FUNCTION GUESN(JJJ,WANT,GC,REST)                                  NUCMOD  1060
      DIMENSION TRY(10),CALS(10)                                        NUCMOD  1061
      CALS(JJJ)=GC                                                      NUCMOD  1062
      TRY(JJJ)=REST                                                     NUCMOD  1063
      IF(JJJ.EQ.1)GO TO 4                                               NUCMOD  1064
      A1=0.                                                             NUCMOD  1065
      A2=0.                                                             NUCMOD  1066
      A3=0.                                                             NUCMOD  1067
      A4=0.                                                             NUCMOD  1068
      ZN=0.                                                             NUCMOD  1069
      DO 2 I=1,JJJ                                                      NUCMOD  1070
      Z=1./(WANT-CALS(I))**2                                            NUCMOD  1071
      A=ALOG(TRY(I))                                                    NUCMOD  1072
      C=ALOG(CALS(I))                                                   NUCMOD  1073
      ZN=ZN+Z                                                           NUCMOD  1074
      A1=A1+A*A*Z                                                       NUCMOD  1075
      A2=A2+A*Z                                                         NUCMOD  1076
      A3=A3+A*C*Z                                                       NUCMOD  1077
    2 A4=A4+C*Z                                                         NUCMOD  1078
      YVAL=(A2*A3-A1*A4)/(A2*A2-A1*ZN)                                  NUCMOD  1079
      SLOPE=(A3-YVAL*A2)/A1                                             NUCMOD  1080
      A=EXP(YVAL)                                                       NUCMOD  1081
      B=SLOPE                                                           NUCMOD  1082
    3 GUESN=(WANT/A)**(1./B)                                            NUCMOD  1083
      RETURN                                                            NUCMOD  1084
    4 B=-2.                                                             NUCMOD  1085
      A=CALS(1)*TRY(1)**2                                               NUCMOD  1086
      GO TO 3                                                           NUCMOD  1087
      END                                                               NUCMOD  1088
```

```
      SUBROUTINE GAMM(BURST,W,HORF,REST,GC,GD,GR)                      NUCMOD   1089
      IF(W.GT.500.0)GO TO 20                                           NUCMOD   1090
      SHOB=BURST/(W**(1./3.))                                          NUCMOD   1091
      IF(SHOB.GT.369.0)GO TO 20                                        NUCMOD   1092
      IF(SHOB.GE.177.0.AND.SHOB.LE.369.0)THSO=6.21*1.0E+06*((W/BURST**3) NUCMOD   1093
     1-(2.0E-08))                                                      NUCMOD   1094
      IF(SHOB.LT.177.0)THSO=1.0                                        NUCMOD   1095
      IF(SHOB.GT.369.0)THSO=0.0                                        NUCMOD   1096
      TSO=1.0-THSO                                                     NUCMOD   1097
      IF(SHOB.LE.183.0)FHW=(3.41E-3*SHOB)+1.0                          NUCMOD   1098
      IF(SHOB.GT.183.0)FHW=1.63                                        NUCMOD   1099
      SINT=(HORF/REST)                                                 NUCMOD   1100
      THETA1=(2.0/3.0)*ASIN(SINT)                                      NUCMOD   1101
      THETA=COS(THETA1)                                                NUCMOD   1102
      FACTOR=((TSO)+(THSO*FHW*THETA))                                  NUCMOD   1103
      GC=GC*FACTOR                                                     NUCMOD   1104
      GR=GR*FACTOR                                                     NUCMOD   1105
      GD=GD*FACTOR                                                     NUCMOD   1106
      GO TO 21                                                         NUCMOD   1107
   20 GC=GC                                                            NUCMOD   1108
      GR=GR                                                            NUCMOD   1109
      GD=GD                                                            NUCMOD   1110
   21 CONTINUE                                                         NUCMOD   1111
      RETURN                                                           NUCMOD   1112
      END                                                              NUCMOD   1113
```

```
      SUBROUTINE AINT (ZF1,ZF2,WAVEL,V,IWDN,PAVE,WIN,SIGAVE,AOACAV,WCO2) NUCMOD   1114
C                                                                       NUCMOD   1115
C     COMPUTES REQUIRED ALTITUDE INTEGRALS                              NUCMOD   1116
C     ZF1 AND ZF2 ARE SOURCE AND RECEIVER ALTITUDES IN FEET.            NUCMOD   1117
C     WAVEL IS THE CENTER WAVE LENGTH IN MICRONS.                       NUCMOD   1118
C     V IS THE SURFACE VISUAL RANGE IN MILES.                           NUCMOD   1119
C     IWDN IS ATMOSPHERE CODE - 1 = WET ATMOSPHERE                      NUCMOD   1120
C                               2 = DRY ATMOSPHERE                      NUCMOD   1121
C     PAVE IS THE RETURNED AVERAGE PRESSURE IN MM-HG.                   NUCMOD   1122
C     WIN IS THE RETURNED WATER INTEGRAL FOR EIGHT DIFFERENT            NUCMOD   1123
C     VALUES OF EXPONENTS OF PRESSURE.                                  NUCMOD   1124
C                                                                       NUCMOD   1125
C     IF ZF1=ZF2, VALUE OF INTEGRAND IS RETURNED IN WIN.                NUCMOD   1126
C     SIGAVE IS THE RETURNED AVERAGE SCATTERING COEFFICIENT IN KM-1.    NUCMOD   1127
C     AOACAV IS THE RETURNED AVERAGE ATMOSPHERIC OZONE ABSORPTION       NUCMOD   1128
C     COEFFICIENT IN KM-1.                                              NUCMOD   1129
C     WCO2 IS THE RETURNED CARBON DIOXIDE CONCENTRATION IN ATMOS-CM/KM. NUCMOD   1130
C                                                                       NUCMOD   1131
      DIMENSION WIN(8), EX(8), WAVEL(14), SIGAVE(14), AOACAV(4), AOAC(4) NUCMOD   1132
C                                                                       NUCMOD   1133
      DATA EX/0.0,0.43,0.52,0.54,0.60,0.61,0.62,0.64/                   NUCMOD   1134
C                                                                       NUCMOD   1135
      IF (ABS(ZF1-ZF2).GT.33.0) GO TO 3                                 NUCMOD   1136
      CALL ATMOS(ZF1,TM,X2X,RHO,THETA,PAVE,CA,AMU,K)                    NUCMOD   1137
      PAVE=760.0*PAVE                                                   NUCMOD   1138
      CALL WATRHO (ZF1,IWDN,RHOW)                                       NUCMOD   1139
      DO 1 I=1,8                                                        NUCMOD   1140
    1 WIN(I)=PAVE**EX(I)*RHOW                                           NUCMOD   1141
      DO 2 IBAND=1,14                                                   NUCMOD   1142
      ALAM=WAVEL(IBAND)                                                 NUCMOD   1143
      CALL SCAT (ZF1,ALAM,V,SIG)                                        NUCMOD   1144
    2 SIGAVE(IBAND)=SIG                                                 NUCMOD   1145
      CALL OZAC (ZF1,AOACAV)                                            NUCMOD   1146
      CALL WCO2P (ZF1,WCO2,0)                                           NUCMOD   1147
      WCO2=-WCO2                                                        NUCMOD   1148
      RETURN                                                            NUCMOD   1149
    3 DZ=ABS(ZF1-ZF2)/20.0                                              NUCMOD   1150
      Z=AMIN1(ZF1,ZF2)-DZ/2.0                                           NUCMOD   1151
      PAVE=0.0                                                          NUCMOD   1152
      DO 4 I=1,8                                                        NUCMOD   1153
    4 WIN(I)=0.0                                                        NUCMOD   1154
      DO 5 IBAND=1,14                                                   NUCMOD   1155
    5 SIGAVE(IBAND)=0.0                                                 NUCMOD   1156
      DO 6 I=1,4                                                        NUCMOD   1157
    6 AOACAV(I)=0.0                                                     NUCMOD   1158
      DO 10 J=1,20                                                      NUCMOD   1159
      Z=Z+DZ                                                            NUCMOD   1160
      CALL ATMOS(Z,TM,X2X,RHO,THETA,P,CA,AMU,K)                         NUCMOD   1161
      P=760.0*P                                                         NUCMOD   1162
      PAVE=PAVE+P                                                       NUCMOD   1163
```

```
      CALL WATRHO (Z,IWDN,RHOW)                                         NUCMOD 1164
      DO 7 I=1,8                                                        NUCMOD 1165
    7 WIN(I)=WIN(I)+P**EX(I)*RHOW                                       NUCMOD 1166
      DO 8 IBAND=1,14                                                   NUCMOD 1167
      ALAM=WAVEL(IBAND)                                                 NUCMOD 1168
      CALL SCAT (Z,ALAM,V,SIG)                                          NUCMOD 1169
    8 SIGAVE(IBAND)=SIGAVE(IBAND)+SIG                                   NUCMOD 1170
      CALL OZAC (Z,AOAC)                                                NUCMOD 1171
      DO 9 I=1,4                                                        NUCMOD 1172
    9 AOACAV(I)=AOACAV(I)+AOAC(I)                                       NUCMOD 1173
   10 CONTINUE                                                          NUCMOD 1174
      CALL WCO2P (ZF1,ARG1,1)                                           NUCMOD 1175
      CALL WCO2P (ZF2,ARG2,1)                                           NUCMOD 1176
      WCO2=(ARG2-ARG1)/((ZF1-ZF2)*0.0003048)                            NUCMOD 1177
      PAVE=PAVE/20.0                                                    NUCMOD 1178
      DO 11 I=1,8                                                       NUCMOD 1179
   11 WIN(I)=WIN(I)/20.0                                                NUCMOD 1180
      DO 12 IBAND=1,14                                                  NUCMOD 1181
   12 SIGAVE(IBAND)=SIGAVE(IBAND)/20.0                                  NUCMOD 1182
      DO 13 I=1,4                                                       NUCMOD 1183
   13 AOACAV(I)=AOACAV(I)/20.0                                          NUCMOD 1184
      RETURN                                                            NUCMOD 1185
      END                                                               NUCMOD 1186
```

```
      FUNCTION FLUX(TT,IMARK)                                           NUCMOD  1187
C                                                                       NUCMOD  1188
C     ****************************************************************** NUCMOD  1189
C                                                                       NUCMOD  1190
C     AFWL THERMAL POWER TIME HISTORY FUNCTION                          NUCMOD  1191
C                                                                       NUCMOD  1192
C     CALLING SEQUENCE                                                  NUCMOD  1193
C                                                                       NUCMOD  1194
C        FLUX(W,SIGMA,TT,FIRST)                                         NUCMOD  1195
C                                                                       NUCMOD  1196
C           W=YIELD (KILOTONS)                                          NUCMOD  1197
C                                                                       NUCMOD  1198
C           SIGMA=DENSITY RATIO AT BURST ALTITUDE (RHOA/RHOSL)          NUCMOD  1199
C                                                                       NUCMOD  1200
C           TT=TIME WHERE POWER IS DESIRED (SEC)                        NUCMOD  1201
C                                                                       NUCMOD  1202
C           FLUX=POWER (WATTS)                                          NUCMOD  1203
C                                                                       NUCMOD  1204
C                                                                       NUCMOD  1205
C     ROUTINE BY ALFRED L. SHARP - 7 NOVEMBER 1972.                     NUCMOD  1206
C                                                                       NUCMOD  1207
C     ****************************************************************** NUCMOD  1208
C                                                                       NUCMOD  1209
      COMMON /SNCM3/    DT,       W,       SIGMA,    TMAX               NUCMOD  1210
C                                                                       NUCMOD  1211
      IF(IMARK.GT.0)GO TO 30                                            NUCMOD  1212
   20 IF (SIGMA.LT.0.0008) GO TO 90                                     NUCMOD  1213
      S=1.75995*SIGMA**(-.0432)                                         NUCMOD  1214
      PSEM=.17214*SIGMA**(-.6244)*W**.53                                NUCMOD  1215
      SIGLOG=ALOG(SIGMA)                                                NUCMOD  1216
      TSEM=(1.6828E-4+SIGLOG*(-7.62326E-6+SIGLOG*(-3.72132E-6-SIGLOG*   NUCMOD  1217
     *1.51933E-7)))*W**.3123                                            NUCMOD  1218
      BMR=2.*SIGMA**.1008                                               NUCMOD  1219
      BML=SIGMA**.1189                                                  NUCMOD  1220
      TMAX=.038*W**.44*SIGMA**(.36)                                     NUCMOD  1221
      TMIN=.00256*W**.39*SIGMA**(-.062)                                 NUCMOD  1222
      DT=(TMAX/TMIN)**.01                                               NUCMOD  1223
      IF (TMIN.GT..99*TMAX) TMIN=.99*TMAX                               NUCMOD  1224
      PMAX=1.49E13*W**.59*SIGMA**(-.45)                                 NUCMOD  1225
      PMIN=6.82E11*W**.54*SIGMA**(-1.02)                                NUCMOD  1226
      IF (PMIN.GE..99*PMAX) PMIN=.99*PMAX                               NUCMOD  1227
      BSEM=PSEM/PMIN-1.                                                 NUCMOD  1228
      BNR=3.4071*SIGMA**.17827                                          NUCMOD  1229
      BNL=9.83382+SIGLOG*(-.45+SIGLOG*(-.207-SIGLOG*.007741))           NUCMOD  1230
      BTIME=1.523*SIGMA**(-.0752)*TMAX                                  NUCMOD  1231
      BHITE=1./(.8*SIGMA**.3372)-1.                                     NUCMOD  1232
      C=(ALOG(TMAX)-ALOG(TMIN))*TMAX/(TMAX-TMIN)                        NUCMOD  1233
      B=S*EXP(C)/C                                                      NUCMOD  1234
      TR=TMIN/TMAX                                                      NUCMOD  1235
      HINTMN=PMIN/(TR**(-S)*EXP(-B*EXP(-C*TR)))                         NUCMOD  1236
```

```
           HINTMA=PMAX/(EXP(-B*EXP(-C)))                                    NUCMOD  1237
           ALPH=.0289*W**.3209+1.023*W**(-.0013979)                         NUCMOD  1238
      C                                                                     NUCMOD  1239
      30   TSMAX=TT/TMAX                                                    NUCMOD  1240
           IF (TT.EQ.0.) GO TO 100                                          NUCMOD  1241
           IF (TT.LT.TMIN) GO TO 50                                         NUCMOD  1242
           IF (TSMAX-1.) 40,60,60                                           NUCMOD  1243
      40   TETA=(1.-ALOG(1./TSMAX)/ALOG(1./TR))                             NUCMOD  1244
           TETAB=3.14159265359*TETA                                         NUCMOD  1245
           HINT=HINTMX+(HINTMN-HINTMX)*((1.+COS(TETAB))/2.)**ALPH           NUCMOD  1246
           GO TO 70                                                         NUCMOD  1247
      50   FAC=2.*BHITE/((BTIME/TT)**BNL+(TT/BTIME)**BNR)                   NUCMOD  1248
           BFAC=1./(FAC+1.)                                                 NUCMOD  1249
           FACM=(2.*BSEM/((TSEM/TT)**BML+(TT/TSEM)**BMR))+1.                NUCMOD  1250
           FLUX=PMIN*BFAC*FACM                                              NUCMOD  1251
           GO TO 80                                                         NUCMOD  1252
      60   HINT=HINTMX                                                      NUCMOD  1253
      70   FAC=2.*BHITE/((BTIME/TT)**BNL+(TT/BTIME)**BNR)                   NUCMOD  1254
           BFAC=1./(FAC+1.)                                                 NUCMOD  1255
           FACM=(2.*BSEM/((TSEM/TT)**BML+(TT/TSEM)**BMR))+1.                NUCMOD  1256
           FLUX=HINT*(TSMAX**(-5)*EXP(-B*EXP(-C*TSMAX)))*BFAC*FACM          NUCMOD  1257
      80   RETURN                                                           NUCMOD  1258
      C                                                                     NUCMOD  1259
        90 CALL NUCERR(6HFLUX  ,1)                                          NUCMOD  1260
           RETURN                                                           NUCMOD  1261
      C                                                                     NUCMOD  1262
      100  FLUX=PMIN                                                        NUCMOD  1263
           RETURN                                                           NUCMOD  1264
           END                                                              NUCMOD  1265
```

```
      SUBROUTINE OZAC (ZF,AOAC)                                         NUCMOD    1266
C                                                                       NUCMOD    1267
C     COMPUTES OZONE ABSORPTION COEFFICIENT IN KM-1                     NUCMOD    1268
C     ZF IS ALTITUDE IN FEET.                                           NUCMOD    1269
C     AOAC IS THE RETURNED ATMOSPHERIC OZONE ABSORPTION COEFFICIENT.    NUCMOD    1270
C                                                                       NUCMOD    1271
      DIMENSION AOAC(4), A(4), HO(26), OT(26)                           NUCMOD    1272
C                                                                       NUCMOD    1273
      DATA A/210.0,101.0,0.034,0.017/,HO/0.0,4.0,6.0,8.0,10.0,11.0,12.0, NUCMOD    1274
     113.0,14.0,15.0,16.0,17.0,18.0,21.0,22.0,23.0,24.0,25.0,30.0,32.0,3 NUCMOD    1275
     24.0,38.0,42.0,45.0,47.0,50.0/,OT/0.00356,0.00226,0.00216,0.00228,0 NUCMOD    1276
     3.0035,0.0046,0.00621,0.00845,0.00957,0.00994,0.0103,0.0111,0.0122, NUCMOD    1277
     40.0184,0.0197,0.0198,0.0193,0.0180,0.00903,0.00682,0.00485,0.00253 NUCMOD    1278
     5,0.00126,0.000919,0.000798,0.000647/                              NUCMOD    1279
C                                                                       NUCMOD    1280
      Z=ZF*0.0003048                                                    NUCMOD    1281
      CALL INTI (Z,26,HO,OT,D)                                          NUCMOD    1282
      DO 1 I=1,4                                                        NUCMOD    1283
    1 AOAC(I)=D*A(I)                                                    NUCMOD    1284
      RETURN                                                            NUCMOD    1285
      END                                                               NUCMOD    1286
```

```
      FUNCTION POWER(TIME,IMARK)                                        NUCMOD    1287
C                                                                       NUCMOD    1288
      COMMON /SNCM3/    DT,        A,        SIGMA,    TMAX             NUCMOD    1289
C                                                                       NUCMOD    1290
      DATA A/1.E-6/                                                     NUCMOD    1291
C                                                                       NUCMOD    1292
      IF(IMARK.GT.0)GO TO 10                                            NUCMOD    1293
      PO = FLUX(A,0)                                                    NUCMOD    1294
      ALDT=ALOG(DT)                                                     NUCMOD    1295
      POPR=.5E-6*PO                                                     NUCMOD    1296
      POWER=0.                                                          NUCMOD    1297
      I=1                                                               NUCMOD    1298
      TSTEP2=TIME/2.                                                    NUCMOD    1299
      RETURN                                                            NUCMOD    1300
C                                                                       NUCMOD    1301
   10 IF(I-1) 1,1,2                                                     NUCMOD    1302
    1 POWER=POPR                                                        NUCMOD    1303
      PLAST=PO                                                          NUCMOD    1304
      TSTART=A                                                          NUCMOD    1305
      I=2                                                               NUCMOD    1306
      GO TO 3                                                           NUCMOD    1307
    2 POWER=0.                                                          NUCMOD    1308
      TSTART=TLAST                                                      NUCMOD    1309
    3 TSTOP=TIME+TSTEP2                                                 NUCMOD    1310
      TLOG=ALOG(TSTOP/TSTART)                                           NUCMOD    1311
      N=TLOG/ALDT                                                       NUCMOD    1312
      IF (N.LT.1) N=1                                                   NUCMOD    1313
      AMULT=EXP(TLOG/FLOAT(N))                                          NUCMOD    1314
      TLAST=TSTART                                                      NUCMOD    1315
      DO 4 J=1,N                                                        NUCMOD    1316
      T=TLAST*AMULT                                                     NUCMOD    1317
      P = FLUX(T,1)                                                     NUCMOD    1318
      POWER=POWER+(P+PLAST)*(T-TLAST)                                   NUCMOD    1319
      TLAST=T                                                           NUCMOD    1320
    4 PLAST=P                                                           NUCMOD    1321
C                                                                       NUCMOD    1322
C     TAKES CARE OF    THE FACTOR OF 2 FOR TRAPEZOIDAL INTEGRATION AND  NUCMOD    1323
C     CONVERTS (WATT-SECONDS) TO ( BTU )                                NUCMOD    1324
C                                                                       NUCMOD    1325
      POWER=0.0004741333*POWER                                          NUCMOD    1326
      RETURN                                                            NUCMOD    1327
      END                                                               NUCMOD    1328
```

```
      SUBROUTINE REFL(S,H,RR)                                            NUCMOD  1329
      DIMENSION XXXX(20),YYYY(21),ZZZZ(21,20)                            NUCMOD  1330
      DATA XXXX/                                                         NUCMOD  1331
     $ 0.          , 1.00000E+00, 2.00000E+00, 3.00000E+00, 4.00000E+00, NUCMOD  1332
     $ 5.00000E+00, 6.00000E+00, 8.00000E+00, 1.00000E+01, 1.20000E+01,  NUCMOD  1333
     $ 1.40000E+01, 1.60000E+01, 1.80000E+01, 2.00000E+01, 2.20000E+01,  NUCMOD  1334
     $ 2.50000E+01, 3.00000E+01, 3.50000E+01, 4.00000E+01, 1.00000E+02/  NUCMOD  1335
      DATAYYYY/                                                          NUCMOD  1336
     $ 0.          , 2.50000E-01, 5.00000E-01, 7.50000E-01, 1.00000E+00, NUCMOD  1337
     $ 1.50000E+00, 2.00000E+00, 3.00000E+00, 4.00000E+00, 6.00000E+00,  NUCMOD  1338
     $ 8.00000E+00, 1.00000E+01, 1.20000E+01, 1.40000E+01, 1.80000E+01,  NUCMOD  1339
     $ 2.20000E+01, 2.60000E+01, 3.00000E+01, 3.50000E+01, 4.00000E+01,  NUCMOD  1340
     $ 1.00000E+02/                                                      NUCMOD  1341
      DATA(ZZZZ(I),I=1,83)/                                              NUCMOD  1342
     $ 1.00000E+00, 4.94300E-01, 1.64300E-01, 4.29000E-02, 1.00000E-03,  NUCMOD  1343
     $ 6.71000E-02, 1.99000E-01, 4.53400E-01, 6.67000E-01, 9.72000E-01,  NUCMOD  1344
     $ 1.16700E+00, 1.31000E+00, 1.41000E+00, 1.48500E+00, 1.59400E+00,  NUCMOD  1345
     $ 1.65900E+00, 1.72200E+00, 1.76400E+00, 1.81000E+00, 1.83500E+00,  NUCMOD  1346
     $ 1.83500E+00,                                                      NUCMOD  1347
     $ 7.07000E-01, 5.37500E-01, 4.19600E-01,  .03400E-01, 2.69300E-01,  NUCMOD  1348
     $ 2.60900E-01, 3.11000E-01, 5.03400E-01, 6.86100E-01, 9.79000E-01,  NUCMOD  1349
     $ 1.15000E+00, 1.30000E+00, 1.40000E+00, 1.48000E+00, 1.59200E+00,  NUCMOD  1350
     $ 1.66800E+00, 1.72000E+00, 1.76200E+00, 1.80600E+00, 1.83200E+00,  NUCMOD  1351
     $ 1.83200E+00,                                                      NUCMOD  1352
     $ 4.47000E-01, 4.31400E-01, 4.36800E-01, 4.30200E-01, 4.37200E-01,  NUCMOD  1353
     $ 4.48000E-01, 4.72400E-01, 4.57100E-01, 7.20600E-01, 9.63000E-01,  NUCMOD  1354
     $ 1.14900E+00, 1.28000E+00, 1.38200E+00, 1.46000E+00, 1.58200E+00,  NUCMOD  1355
     $ 1.66000E+00, 1.71200E+00, 1.75600E+00, 1.79900E+00, 1.82300E+00,  NUCMOD  1356
     $ 1.82300E+00,                                                      NUCMOD  1357
     $ 3.16000E-01, 3.22600E-01, 3.43500E-01, 3.72400E-01, 4.00400E-01,  NUCMOD  1358
     $ 4.56300E-01, 5.08500E-01, 6.17100E-01, 7.31000E-01, 9.42500E-01,  NUCMOD  1359
     $ 1.12300E+00, 1.25800E+00, 1.35000E+00, 1.44000E+00, 1.56600E+00,  NUCMOD  1360
     $ 1.64600E+00, 1.70200E+00, 1.74900E+00, 1.79000E+00, 1.81900E+00/  NUCMOD  1361
      DATA(ZZZZ(I),I=84,162)/                                            NUCMOD  1362
     $ 1.81900E+00,                                                      NUCMOD  1363
     $ 2.43000E-01, 2.49000E-01, 2.70000E-01, 2.96000E-01, 3.29000E-01,  NUCMOD  1364
     $ 3.96700E-01, 4.62500E-01, 5.84800E-01, 7.03700E-01, 9.08200E-01,  NUCMOD  1365
     $ 1.05600E+00, 1.19100E+00, 1.32600E+00, 1.41000E+00, 1.54200E+00,  NUCMOD  1366
     $ 1.64000E+00, 1.68500E+00, 1.73700E+00, 1.78000E+00, 1.81000E+00,  NUCMOD  1367
     $ 1.81000E+00,                                                      NUCMOD  1368
     $ 1.96200E-01, 2.00000E-01, 2.17700E-01, 2.36000E-01, 2.67200E-01,  NUCMOD  1369
     $ 3.26500E-01, 3.95700E-01, 5.25300E-01, 6.35000E-01, 8.60000E-01,  NUCMOD  1370
     $ 1.03800E+00, 1.18000E+00, 1.29200E+00, 1.38100E+00, 1.52800E+00,  NUCMOD  1371
     $ 1.61000E+00, 1.67200E+00, 1.72200E+00, 1.77000E+00, 1.79500E+00,  NUCMOD  1372
     $ 1.79500E+00,                                                      NUCMOD  1373
     $ 1.64400E-01, 1.71000E-01, 1.80600E-01, 2.00000E-01, 2.20100E-01,  NUCMOD  1374
     $ 2.73700E-01, 3.34400E-01, 4.60500E-01, 5.72200E-01, 8.00000E-01,  NUCMOD  1375
     $ 9.81100E-01, 1.12700E+00, 1.24100E+00, 1.34100E+00, 1.48800E+00,  NUCMOD  1376
     $ 1.57900E+00, 1.64900E+00, 1.70600E+00, 1.75300E+00, 1.78400E+00,  NUCMOD  1377
     $ 1.78400E+00,                                                      NUCMOD  1378
```

```
     $ 1.24000E-01, 1.29000E-01, 1.34000E-01, 1.46000E-01, 1.59900E-01,  NUCMOD   1379
     $ 1.96500E-01, 2.42900E-01, 3.44600E-01, 4.55000E-01, 6.67400E-01,  NUCMOD   1380
     $ 8.52700E-01, 1.00600E+00, 1.13300E+00, 1.24600E+00, 1.40900E+00/  NUCMOD   1381
      DATA(ZZZZ(I),I=163,241)/                                           NUCMOD   1382
     $ 1.52000E+00, 1.60000E+00, 1.66300E+00, 1.72100E+00, 1.76000E+00,  NUCMOD   1383
     $ 1.76000E+00,                                                      NUCMOD   1384
     $ 9.95000E-02, 1.00000E-01, 1.06500E-01, 1.15000E-01, 1.23700E-01,  NUCMOD   1385
     $ 1.48500E-01, 1.82000E-01, 2.62000E-01, 3.53600E-01, 5.45700E-01,  NUCMOD   1386
     $ 7.2590 E-01, 8.84100E-01, 1.02200E+00, 1.14000E+00, 1.32100E+00,  NUCMOD   1387
     $ 1.45100E+00, 1.54700E+00, 1.62100E+00, 1.69000E+00, 1.73500E+00,  NUCMOD   1388
     $ 1.73500E+00,                                                      NUCMOD   1389
     $ 8.30000E-02, 8.50000E-02, 8.90000E-02, 9.10000E-02, 1.00000E-01,  NUCMOD   1390
     $ 1.17000E-01, 1.43000E-01, 2.05000E-01, 2.78000E-01, 4.41000E-01,  NUCMOD   1391
     $ 6.10000E-01, 7.68000E-01, 9.08000E-01, 1.02800E+00, 1.22800E+00,  NUCMOD   1392
     $ 1.37000E+00, 1.47800E+00, 1.56200E+00, 1.64900E+00, 1.70100E+00,  NUCMOD   1393
     $ 1.77100E+00,                                                      NUCMOD   1394
     $ 7.12000E-02, 7.20000E-02, 7.56000E-02, 8.00000E-02, 8.40000E-02,  NUCMOD   1395
     $ 9.82000E-02, 1.16600E-01, 1.64700E-01, 2.24900E-01, 3.66400E-01,  NUCMOD   1396
     $ 5.17300E-01, 6.65400E-01, 8.00000E-01, 9.24400E-01, 1.13100E+00,  NUCMOD   1397
     $ 1.28800E+00, 1.40100E+00, 1.50300E+00, 1.60000E+00, 1.66300E+00,  NUCMOD   1398
     $ 1.663E+00,                                                        NUCMOD   1399
     $ 6.24000E-02, 6.30000E-02, 6.50000E-02, 6.90000E-02, 7.20000E-02,  NUCMOD   1400
     $ 8.40000E-02, 9.80000E-02, 1.38000E-01, 1.85000E-01, 3.07000E-01/  NUCMOD   1401
      DATA(ZZZZ(I),I=242,320)/                                           NUCMOD   1402
     $ 4.39000E-01, 5.71000E-01, 7.06000E-01, 8.30000E-01, 1.04000E+00,  NUCMOD   1403
     $ 1.20300E+00, 1.32900E+00, 1.43200E+00, 1.54300E+00, 1.61700E+00,  NUCMOD   1404
     $ 1.61700E+00,                                                      NUCMOD   1405
     $ 5.55000E-02, 5.60000E-02, 5.67600E-02, 6.00000E-02, 6.43000E-02,  NUCMOD   1406
     $ 7.23000E-02, 8.38000E-02, 1.15300E-01, 1.55600E-01, 2.60000E-01,  NUCMOD   1407
     $ 3.74500E-01, 4.98000E-01, 6.20000E-01, 7.383E0E-01, 9.46700E-01,  NUCMOD   1408
     $ 1.12000E+00, 1.25200E+00, 1.37000E+00, 1.48100E+00, 1.56700E+00,  NUCMOD   1409
     $ 1.56700E+00,                                                      NUCMOD   1410
     $ 4.95000E-02, 5.00000E-02, 5.05000E-02, 5.20000E-02, 5.65000E-02,  NUCMOD   1411
     $ 6.90000E-02, 7.25000E-02, 9.90000E-02, 1.31000E-01, 2.19000E-01,  NUCMOD   1412
     $ 3.21000E-01, 4.32000E-01, 5.50000E-01, 6.56000E-01, 8.61000E-01,  NUCMOD   1413
     $ 1.04000E+00, 1.18100E+00, 1.30000E+00, 1.43700E+00, 1.51600E+00,  NUCMOD   1414
     $ 1.51600E+00,                                                      NUCMOD   1415
     $ 4.54000E-02, 4.54000E-02, 4.54000E-02, 4.90000E-02, 5.08000E-02,  NUCMOD   1416
     $ 5.65000E-02, 6.46000E-02, 8.62000E-02, 1.14800E-01, 1.89500E-01,  NUCMOD   1417
     $ 2.79900E-01, 3.80200E-01, 4.88000E-01, 5.89800E-01, 7.87400E-01,  NUCMOD   1418
     $ 9.61400E-01, 1.11800E+00, 1.23400E+00, 1.35900E+00, 1.46300E+00,  NUCMOD   1419
     $ 1.46300E+00,                                                      NUCMOD   1420
     $ 3.84000E-02, 3.90000E-02, 4.00000E-02, 4.10000E-02, 4.30000E-02/  NUCMOD   1421
      DATA(ZZZZ(I),I=321,403)/                                           NUCMOD   1422
     $ 4.60000E-02, 5.60000E-02, 6.70000E-02, 8.90000E-02, 1.42000E-01,  NUCMOD   1423
     $ 2.18000E-01, 2.99000E-01, 3.88000E-01, 4.82000E-01, 6.52000E-01,  NUCMOD   1424
     $ 8.26000E-01, 9.83000E-01, 1.10900E+00, 1.23800E+00, 1.35200E+00,  NUCMOD   1425
     $ 1.35200E+00,                                                      NUCMOD   1426
     $ 3.33000E-02, 3.30000E-02, 3.30000E-02, 3.40000E-02, 3.57000E-02,  NUCMOD   1427
     $ 3.93000E-02, 4.37000E-02, 5.56000E-02, 7.17000E-02, 1.15200E-01,  NUCMOD   1428
```

```
      $ 1.71400E-01, 2.37200E-01, 3.08000E-01, 3.88100E-01, 5.46900E-01,  NUCMOD  1429
      $ 7.04200E-01, 8.52000E-01, 9.81500E-01, 1.12300E+00, 1.24100E+00,  NUCMOD  1430
      $ 1.24100E+00,                                                      NUCMOD  1431
      $ 2.86000E-02, 2.80000E-02, 2.80000E-02, 2.80000E-02, 2.90000E-02,  NUCMOD  1432
      $ 3.30000E-02, 3.60000E-02, 4.10000E-02, 5.80000E-02, 9.00000E-02,  NUCMOD  1433
      $ 1.32000E-01, 1.90000E-01, 2.52000E-01, 3.18000E-01, 4.57000E-01,  NUCMOD  1434
      $ 6.03000E-01, 7.43000E-01, 8.48000E-01, 9.96000E-01, 1.11800E+00,  NUCMOD  1435
      $ 1.11800E+00,                                                      NUCMOD  1436
      $ 2.50000E-02, 2.50000E-02, 2.50000E-02, 2.50000E-02, 2.61000E-02,  NUCMOD  1437
      $ 2.82000E-02, 3.08000E-02, 3.76000E-02, 4.70000E-02, 7.24000E-02,  NUCMOD  1438
      $ 1.06400E-01, 1.47700E-01, 1.92000E-01, 2.47900E-01, 3.63000E-01,  NUCMOD  1439
      $ 4.85500E-01, 6.09000E-01, 7.29300E-01, 8.70000E-01, 1.00100E+00,  NUCMOD  1440
      $ 1.00100E+00, 2.50000E-02, 2.50000E-02, 2.50000E-02, 2.50000E-02/  NUCMOD  1441
       DATA(ZZZZ(I),I=404,420)/                                           NUCMOD  1442
      $ 2.61000E-02,                                                      NUCMOD  1443
      $ 2.82000E-02, 3.08000E-02, 3.76000E-02, 4.70000E-02, 7.24000E-02,  NUCMOD  1444
      $ 1.06400E-01, 1.47700E-01, 1.92000E-01, 2.47900E-01, 3.63000E-01,  NUCMOD  1445
      $ 4.85500E-01, 6.09000E-01, 7.29300E-01, 8.70000E-01, 1.00100E+00,  NUCMOD  1446
      $ 1.00100E+00/                                                      NUCMOD  1447
C                                                                         NUCMOD  1448
       X1=G                                                               NUCMOD  1449
       IF(X1.GT.XXXX(19))X1=XXXX(19)                                      NUCMOD  1450
       IF (X1 .LT. XXXX(1)) X1=XXXX(1)                                    NUCMOD  1451
       X2=H                                                               NUCMOD  1452
       IF(X2.GT.YYYY(20))X2=YYYY(20)                                      NUCMOD  1453
       IF (X2 .LT. YYYY(1)) X2=YYYY(1)                                    NUCMOD  1454
C                                                                         NUCMOD  1455
       CALL DINT(XXXX,YYYY,ZZZZ,21,20,X1,X2,RR)                           NUCMOD  1456
C                                                                         NUCMOD  1457
       RETURN                                                             NUCMOD  1458
       END                                                                NUCMOD  1459
```

```
      SUBROUTINE DINT (XX,YY,ZZ,M,N,X,Y,Z)                                NUCMOD  1460
C                                                                         NUCMOD  1461
C     LINEAR 2 DIMENSIONAL INTERPOLATION                                  NUCMOD  1462
C     EXAMPLE-GIVEN THE FOLLOWING TABLE,AND X AND Y VALUES,WE CAN         NUCMOD  1463
C     IINTERPOLATE FOR APPROPIATE Z VALUE                                 NUCMOD  1464
C     XX AND YY MUST BE IN INCREASING ORDER                               NUCMOD  1465
C        X1  X2  X3                                                       NUCMOD  1466
C     1Y1 Z11 Z12 Z13                                                     NUCMOD  1467
C     2Y2 Z21 Z22 Z23                                                     NUCMOD  1468
C     3Y3 Z31 Z32 Z33                                                     NUCMOD  1469
      DIMENSION XX(N), YY(M), ZZ(M,N)                                     NUCMOD  1470
      IF (((X.LT.XX(1)).OR.(X.GT.XX(N))).OR.((Y.LT.YY(1)).OR.(Y.GT.YY(M)  NUCMOD  1471
     1))) GO TO 70                                                        NUCMOD  1472
      MM=M-1                                                              NUCMOD  1473
      NN=N-1                                                              NUCMOD  1474
      DO 20 I=1,MM                                                        NUCMOD  1475
      IF ((YY(I).LE.Y).AND.(YY(I+1).GT.Y)) GO TO 10                       NUCMOD  1476
      GO TO 20                                                            NUCMOD  1477
   10 C1=(Y-YY(I))/(YY(I+1)-YY(I))                                        NUCMOD  1478
      I1=I                                                                NUCMOD  1479
      I2=I+1                                                              NUCMOD  1480
      GO TO 30                                                            NUCMOD  1481
   20 CONTINUE                                                            NUCMOD  1482
   30 DO 50 J=1,NN                                                        NUCMOD  1483
      IF ((XX(J).LE.X).AND.(XX(J+1).GT.X)) GO TO 40                       NUCMOD  1484
      GO TO 50                                                            NUCMOD  1485
   40 C2=(X-XX(J))/(XX(J+1)-XX(J))                                        NUCMOD  1486
      J1=J                                                                NUCMOD  1487
      J2=J+1                                                              NUCMOD  1488
      GO TO 60                                                            NUCMOD  1489
   50 CONTINUE                                                            NUCMOD  1490
   60 CONTINUE                                                            NUCMOD  1491
      Z1=ZZ(I1,J1)+C1*(ZZ(I2,J1)-ZZ(I1,J1))                               NUCMOD  1492
      Z2=ZZ(I1,J2)+C1*(ZZ(I2,J2)-ZZ(I1,J2))                               NUCMOD  1493
      Z=Z1+C2*(Z2-Z1)                                                     NUCMOD  1494
      RETURN                                                              NUCMOD  1495
   70 WRITE (6,80) X,Y                                                    NUCMOD  1496
      CALL NUCERR(6HDINT  ,1)                                             NUCMOD  1497
      RETURN                                                              NUCMOD  1498
C                                                                         NUCMOD  1499
C                                                                         NUCMOD  1500
   80 FORMAT (1H1,26HPARAMETER OUTSIDE RANGE,X=,E15.8,2X,2HY=,E15.8)      NUCMOD  1501
      END                                                                 NUCMOD  1502
```

```
      SUBROUTINE SCAT (ZF,ALAM,V,SIG)                                   NUCMOD  1503
C                                                                       NUCMOD  1504
C     SUBROUTINE SCAT COMPUTES THE SCATTERING COEFFICIENT, KM-1.        NUCMOD  1505
C     ZF IS THE ALTITUDE IN FEET.                                       NUCMOD  1506
C     ALAM IS THE WAVELENGTH IN MICRONS.                                NUCMOD  1507
C     V IS THE SURFACE VISUAL RANGE IN MILES.                           NUCMOD  1508
C     SIG IS THE RETURNED SCATTERING COEFFICIENT, KM-1.                 NUCMOD  1509
C                                                                       NUCMOD  1510
      DIMENSION AS(4), AL(4), ES(4), EL(4), VIS(4)                      NUCMOD  1511
C                                                                       NUCMOD  1512
      DATA AS/1.22,0.244,0.0486,0.0152/                                 NUCMOD  1513
      DATA AL/0.168,0.0777,0.0361,0.0152/                               NUCMOD  1514
      DATA ES/-0.833,-0.529,-0.221,-0.0926/                             NUCMOD  1515
      DATA EL/-0.173,-0.147,-0.122,-0.0926/                             NUCMOD  1516
      DATA VIS/2.0,10.0,50.0,160.0/                                     NUCMOD  1517
C                                                                       NUCMOD  1518
      V=AMAX1(V,2.0)                                                    NUCMOD  1519
      V=AMIN1(V,160.0)                                                  NUCMOD  1520
      Z=ZF*0.0003048                                                    NUCMOD  1521
      PWR=0.11*Z+0.70                                                   NUCMOD  1522
      ARG=(0.55/ALAM)**PWR                                              NUCMOD  1523
      DO 1 I=1,2                                                        NUCMOD  1524
      IF (V.LE.VIS(I+1)) GO TO 2                                        NUCMOD  1525
    1 CONTINUE                                                          NUCMOD  1526
      I=3                                                               NUCMOD  1527
    2 IF (Z.GT.3.0) GO TO 3                                             NUCMOD  1528
      SIG1=AS(I)*ARG*EXP(ES(I)*Z)                                       NUCMOD  1529
      SIG2=AS(I+1)*ARG*EXP(ES(I+1)*Z)                                   NUCMOD  1530
      GO TO 4                                                           NUCMOD  1531
    3 SIG1=AL(I)*ARG*EXP(EL(I)*Z)                                       NUCMOD  1532
      SIG2=AL(I+1)*ARG*EXP(EL(I+1)*Z)                                   NUCMOD  1533
    4 SIG=SIG1+(SIG2-SIG1)*(V-VIS(I))/(VIS(I+1)-VIS(I))                 NUCMOD  1534
      RETURN                                                            NUCMOD  1535
      END                                                               NUCMOD  1536
```

```
      SUBROUTINE TCO2(IFLAG)                                            NUCMOD  1537
C                                                                       NUCMOD  1538
      COMMON /SNCM2/     ADACAV(14),     DKM,           SIGAVE(14),     NUCMOD  1539
     *   TCOW(14),       TDIF(14),       TOZ(14),       TRWL(14),       NUCMOD  1540
     *   TWATW(14),      WCO2,           WIN(8)                         NUCMOD  1541
C                                                                       NUCMOD  1542
      COMMON /SNCM4/     PAVE                                           NUCMOD  1543
C                                                                       NUCMOD  1544
      DIMENSION          C(4), XK(4), ALIM(4), CC(4), DELNU(4), XKK(4), NUCMOD  1545
     1 C1(4), IUSE(4), C2(4), C3(4), C4(4), D(4)                        NUCMOD  1546
C                                                                       NUCMOD  1547
      DATA IUSE/9,10,11,13/,CONST/.43429448/                            NUCMOD  1548
      DATA C/.058,.063,.492,3.15/                                       NUCMOD  1549
      DATA XK/.41,.38,.39,.43/                                          NUCMOD  1550
      DATA ALIM/3*80.,50./                                              NUCMOD  1551
      DATA CC/0.,0.,-536.,-137./                                        NUCMOD  1552
      DATA DELNU/1500.,550.,1100.,1060./                                NUCMOD  1553
      DATA XKK/0.,0.,114.,68./                                          NUCMOD  1554
      DATA D/0.,0.,138.,77./                                            NUCMOD  1555
C                                                                       NUCMOD  1556
      IF(IFLAG .GT. 0) GO TO 10                                         NUCMOD  1557
      PALOG=ALOG(PAVE)                                                  NUCMOD  1558
      DO 1 I=1,4                                                        NUCMOD  1559
      C1(I)=C(I)*EXP(PALOG*XK(I))                                       NUCMOD  1560
      C2(I)=D(I)*CONST                                                  NUCMOD  1561
      C3(I)=XKK(I)*CONST*PALOG+CC(I)                                    NUCMOD  1562
      C4(I)=1./DELNU(I)                                                 NUCMOD  1563
    1 CONTINUE                                                          NUCMOD  1564
      RETURN                                                            NUCMOD  1565
C                                                                       NUCMOD  1566
   10 CONTINUE                                                          NUCMOD  1567
      DO 6 I=1,4                                                        NUCMOD  1568
      WWW=WCO2*DKM                                                      NUCMOD  1569
      WSK=SQRT(WWW)                                                     NUCMOD  1570
      AIN=C1(I)*WSK                                                     NUCMOD  1571
      IF (AIN.LE.ALIM(I)) GO TO 3                                       NUCMOD  1572
      IF (D(I)) 3,3,2                                                   NUCMOD  1573
    2 AIN=C2(I)*ALOG(WWW)+C3(I)                                         NUCMOD  1574
      AIN=AMAX1(AIN,ALIM(I))                                            NUCMOD  1575
    3 TCO=1.-AIN*C4(I)                                                  NUCMOD  1576
      IF (TCO) 4,5,5                                                    NUCMOD  1577
    4 TCO=0.                                                            NUCMOD  1578
    5 J=IUSE(I)                                                         NUCMOD  1579
      TCOW(J)=TCO                                                       NUCMOD  1580
    6 CONTINUE                                                          NUCMOD  1581
      RETURN                                                            NUCMOD  1582
      END                                                               NUCMOD  1583
```

```
      SUBROUTINE TDIFFS(IFLAG)                                          NUCMOD   1584
C                                                                       NUCMOD   1585
C     SUBROUTINE TDIFFS COMPUTES THE DIFFUSE SCATTERING FUNCTION, TDIFF.NUCMOD   1586
C     RF IS THE RANGE FROM THE SOURCE TO THE RECEIVER IN FEET.          NUCMOD   1587
C     SIGAVE IS THE SCATTERING COEFFICIENT, KM-1.                       NUCMOD   1588
C     TDIF IS THE RETURNED DIFFUSE SCATTERING FUNCTION.                 NUCMOD   1589
C                                                                       NUCMOD   1590
      COMMON /SNCM2/    ADACAV(14),     DKM,           SIGAVE(14),      NUCMOD   1591
     *   TCOW(14),      TDIF(14),       TOZ(14),       TRWL(14),        NUCMOD   1592
     *   TWATW(14),     WCO2,           WIN(8)                          NUCMOD   1593
C                                                                       NUCMOD   1594
      DIMENSION SMCBRT(14),P6SQRT(14)                                   NUCMOD   1595
C                                                                       NUCMOD   1596
      IF(IFLAG .GT. 0) GO TO 10                                         NUCMOD   1597
      DO 1 IBAND=1,14                                                   NUCMOD   1598
      S=SIGAVE(IBAND)                                                   NUCMOD   1599
      SMCBRT(IBAND)=S-S**.3333333333                                    NUCMOD   1600
    1 P6SQRT(IBAND)=0.6*SQRT(S)                                         NUCMOD   1601
      RETURN                                                            NUCMOD   1602
C                                                                       NUCMOD   1603
   10 CONTINUE                                                          NUCMOD   1604
      DO 2 IBAND=1,14                                                   NUCMOD   1605
    2 TDIF(IBAND)=EXP(-SIGAVE(IBAND)*DKM)*(1.+P6SQRT(IBAND)*DKM*EXP(SMCBNUCMOD   1606
     *RT(IBAND)*DKM))                                                   NUCMOD   1607
      RETURN                                                            NUCMOD   1608
      END                                                               NUCMOD   1609
```

```
      SUBROUTINE TOZONE                                                 NUCMOD   1610
C                                                                       NUCMOD   1611
C     COMPUTES OZONE ABSORPTION FACTOR.                                 NUCMOD   1612
C     RF IS THE RANGE FROM THE SOURCE TO THE RECEIVER IN FEET.          NUCMOD   1613
C     AOACAV IS THE OZONE ABSORPTION COEFFICIENT IN KM-1.               NUCMOD   1614
C     TOZ IS THE RETURNED OZONE TRANSMITTANCE FACTOR.                   NUCMOD   1615
C                                                                       NUCMOD   1616
      COMMON /SNCM2/    AOACAV(14),     DKM,            SIGAVE(14),     NUCMOD   1617
     *   TCOW(14),      TDIF(14),       TOZ(14),        THWL(14),       NUCMOD   1618
     *   TWATW(14),     WCO2,           WIN(8)                          NUCMOD   1619
C                                                                       NUCMOD   1620
      DO 1 I=1,4                                                        NUCMOD   1621
    1 TOZ(I)=EXP(-AOACAV(I)*DKM)                                        NUCMOD   1622
      RETURN                                                            NUCMOD   1623
      END                                                               NUCMOD   1624
```

```
      SUBROUTINE TRANS(DFB)
C
C     SUBROUTINE TRANS CALCULATES THE ATMOSPHERIC
C     TRANSMISSION FACTOR AS A FUNCTION OF WAVELENGTH.
C
      COMMON /SNCM2/    AUACAV(14),    DKM,          SIGAVE(14),
     *   TCOW(14),      TDIF(14),      TOZ(14),      TRWL(14),
     *   TWATW(14),     WCO2,          WIN(8)
C
      DATA TOZ,TWATW,TCOW/42*1.0/
C
      DKM=0.0003048*DFB
      CALL TDIFFS(1)
      CALL TWATER(1)
      CALL TCO2(1)
      CALL TOZONE
      DO 1 I=1,14
      TRWL(I)=TDIF(I)*TWATW(I)*TCOW(I)*TOZ(I)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TWATER(IFLAG)
C
      COMMON /SNCM2/    AOACAV(14),    DKM,           SIGAVE(14),
     *  TCOW(14),       TDIF(14),      TOZ(14),       TRWL(14),
     *  TWATW(14),      WCO2,          WIN(8)
C
      DIMENSION                   IUSE(7), ALIM(7), WINW(7), WINS(7), D
     1ELNU(7), D(7), C(7), CC(7), C1(7), C2(7), C3(7), C4(7)
C
      DATA CONST/0.43429448/,IUSE/4,6,7,9,11,13,14/
      DATA ALIM/0.,200.,200.,350.,275.,200.,500./
      DATA WINW,WINS/14*0./
      DATA DELNU/0.,2240.,1060.,1500.,1100.,1060.,540./
      DATA D/0.,0.,0.,460.,232.,246.,0./
      DATA C/0.,38.,31.,242.,152.,316.,40.2/
      DATA CC/0.,0.,0.,202.,127.,337.,0./
C
      IF(IFLAG .GT. 0) GO TO 10
      WINW(2)=WIN(4)
      WINW(3)=WIN(3)
      WINW(4)=WIN(2)
      WINW(5)=WIN(5)
      WINW(6)=WIN(8)
      WINW(7)=WIN(5)
      WINS(4)=WIN(2)
      WINS(5)=WIN(7)
      WINS(6)=WIN(6)
      C1(1)=0.1*WIN(1)
      DO 1 I=2,7
      C1(I)=C(I)*SQRT(0.1*WINW(I))
    1 C4(I)=1./DELNU(I)
      DO 2 I=4,6
      C2(I)=0.1*WINS(I)
    2 C3(I)=D(I)*CONST
      RETURN
   10 CONTINUE
      DSQRT=SQRT(DKM)
      W=C1(1)*DKM
      TWATW(4)=AMAX1(0.,1.-.02*W)
      DO 5 I=2,7
      AIN=C1(I)*DSQRT
      IF (AIN.LE.ALIM(I)) GO TO 4
      IF (D(I)) 3,4,3
    3 UE=C2(I)*DKM
      AIN=CC(I)+C3(I)*ALOG(UE)
      AIN=AMAX1(ALIM(I),AIN)
    4 TWAT=1.-AIN*C4(I)
      TWAT=AMAX1(0.,TWAT)
      J=IUSE(I)
    5 TWATW(J)=AMIN1(1.0,TWAT)
```

```
 1695        RETURN                                                      NUCMOD   1696
 1696        END                                                         NUCMOD   1697
```

```
      SUBROUTINE VIEWF(VVVF,FBFLAG)                                     NUCMOD  1698
C                                                                       NUCMOD  1699
C     VIEWF CALCULATES THE VIEW FACTOR (RATIO OF NORMAL INCIDENT        NUCMOD  1700
C     RADIATION TO RADIATION INCIDENT OF PLANE NORMAL TO LINE OF SIGHT  NUCMOD  1701
C     TO POINT SOURCE) FOR EACH SURFACE ELEMENT IN SECTION.             NUCMOD  1702
C     VIEWF ALSO CALCULATES GG AND AXR FOR USE IN CALCULATION OF        NUCMOD  1703
C     GROUND AND CLOUD REFLECTIONS.                                     NUCMOD  1704
C                                                                       NUCMOD  1705
      COMMON /SNCM1/     AA,       AB,       AC,       AG,       ALT,   NUCMOD  1706
     *        AXR,      CALT,     CREFL,    DFB,      FBALT,    GALT,   NUCMOD  1707
     *        GG,       HFB,      ICLOUD,   NWL,      G,        GRR(14), NUCMOD  1708
     *        RFB,      TC,       TRAD,     VF,       XFB,      YFB,    NUCMOD  1709
     *        ZFB,      XNA,      YNA,      ZNA                         NUCMOD  1710
C                                                                       NUCMOD  1711
      DIMENSION DBT(10),ALFT(19),VFRT(190)                              NUCMOD  1712
C                                                                       NUCMOD  1713
      DATA ALFT/0.0,5.0,10.0,15.0,20.0,25.0,30.0,35.0,40.0,45.0,50.0,55. NUCMOD  1714
     10,60.0,65.0,70.0,75.0,80.0,85.0,90.0/                             NUCMOD  1715
      DATA DBT/1.0,1.1,1.2,1.5,2.0,3.0,5.0,10.0,100.0,10.0E20/,VFRT/19*1 NUCMOD  1716
     1.0,14*1.0,0.999,0.992,0.966,0.863,0.500,13*1.0,0.996,0.985,0.956,0 NUCMOD  1717
     2.890,0.747,0.500,10*1.0,0.997,0.990,0.975,0.949,0.908,0.844,0.754, NUCMOD  1718
     30.636,0.500,7*1.0,0.999,0.995,0.987,0.973,0.951,0.920,0.878,0.824, NUCMOD  1719
     40.758,0.680,0.593,0.500,5*1.0,0.999,0.995,0.986,0.972,0.953,0.928, NUCMOD  1720
     50.896,0.857,0.811,0.759,0.700,0.637,0.569,0.500,4*1.0,0.996,0.989, NUCMOD  1721
     60.979,0.962,0.942,0.916,0.885,0.850,0.811,0.766,0.718,0.667,0.613, NUCMOD  1722
     70.557,0.500,2*1.0,0.999,0.995,0.987,0.975,0.958,0.938,0.914,0.885, NUCMOD  1723
     80.853,0.815,0.779,0.737,0.693,0.647,0.599,0.550,0.500,1.0,0.999,0. NUCMOD  1724
     9993,0.985,0.972,0.956,0.936,0.913,0.886,0.857,0.825,0.790,0.753,0. NUCMOD  1725
     $714,0.673,0.631,0.588,0.544,0.500,1.0,0.998,0.992,0.983,0.970,0.95 NUCMOD  1726
     $3,0.933,0.910,0.883,0.854,0.821,0.787,0.750,0.711,0.671,0.629,0.58 NUCMOD  1727
     $7,0.544,0.500/                                                    NUCMOD  1728
      DATA NDB/10/,NALF/19/                                             NUCMOD  1729
      DATA PI/3.14159265359/                                            NUCMOD  1730
      DATA PIO2/1.57079633/                                             NUCMOD  1731
C                                                                       NUCMOD  1732
      FBFLAG=0.                                                         NUCMOD  1733
      VF=0.                                                             NUCMOD  1734
      VVVF=0.                                                           NUCMOD  1735
      DX=XFB                                                            NUCMOD  1736
      DY=YFB                                                            NUCMOD  1737
      DZ=ZFB                                                            NUCMOD  1738
      GG=SQRT(DX**2+DY**2)                                              NUCMOD  1739
      DFB=SQRT(DX**2+DY**2+DZ**2)                                       NUCMOD  1740
      IF (ICLOUD.EQ.0) GO TO 1                                          NUCMOD  1741
      IF ((CALT-ALT)*(CALT-FBALT).GT.0.0) GO TO 1                       NUCMOD  1742
      RETURN                                                            NUCMOD  1743
    1 DX=DX/DFB                                                         NUCMOD  1744
      DY=DY/DFB                                                         NUCMOD  1745
      DZ=DZ/DFB                                                         NUCMOD  1746
      VFR=1.0                                                           NUCMOD  1747
```

```
      IF (RFB.LE.HFB) GO TO 3
      VFRO=0.5
      IF (DZ.GE.0.0) GO TO 2
      ALFA=ATAN(SQRT(DX**2+DY**2)/(-DZ))*57.29578
      DBAR=DFB/RFB
      CALL INT2(DBAR,ALFA,NDB,NALF,DBT,ALFT,VFRT,VFRO,190)
    2 VFR=VFRO+(1.0-VFRO)*SIN(PIO2*HFB/RFB)
    3 IF(DFB .GT. RFB) GO TO 4
      FBFLAG=1.
      RETURN
    4 SINB=RFB/DFB
      COST=DX*XNA+DY*YNA+DZ*ZNA
      IF (COST.LT.-SINB) GO TO 7
      IF (COST.LT.SINB) GO TO 5
      VF=COST
      GO TO 6
    5 COSB=SQRT(1.0-SINB**2)
      SINT=SQRT(1.0-COST**2)
      VU=COSB*COST/(SINB*SINT)
      VV=COSB/SINT
      VF=(COST*(PI-ACOS(VU)+VU*SQRT(1.0-VU**2))+(ACOS(VV)-VV*SQRT(1.0-VV
     1**2))/SINB**2)/PI
    6 VF=VF*VFR
    7 CONTINUE
      VVVF=VF
      RETURN
      END
```

```
      SUBROUTINE WATRHO (ZF,IWDN,RHOW)                                  NUCMOD  1775
C                                                                       NUCMOD  1776
C     SUBROUTINE WATRHO COMPUTES WATER VAPOR DENSITY  GM/M**3           NUCMOD  1777
C     ZF IS ALTITUDE IN FEET.                                           NUCMOD  1778
C     IWDN IS ATMOSPHERE CODE.  1 = WET ATMOSPHERE.                     NUCMOD  1779
C                               2 = DRY ATMOSPHERE.                     NUCMOD  1780
C     RHOW IS RETURNED WATER VAPOR DENSITY, GM/M**3                     NUCMOD  1781
C                                                                       NUCMOD  1782
      Z=ZF*0.0003048                                                    NUCMOD  1783
      IF (IWDN.EQ.2) GO TO 3                                            NUCMOD  1784
      IF (Z.GT.4.0) GO TO 1                                             NUCMOD  1785
      RHOW=EXP(3.135-0.3277*Z)                                          NUCMOD  1786
      GO TO 6                                                           NUCMOD  1787
    1 IF (Z.GT.16.7) GO TO 2                                            NUCMOD  1788
      RHOW=EXP(3.962-0.5344*Z)                                          NUCMOD  1789
      GO TO 6                                                           NUCMOD  1790
    2 RHOW=0.007                                                        NUCMOD  1791
      GO TO 6                                                           NUCMOD  1792
    3 IF (Z.GT.5.0) GO TO 4                                             NUCMOD  1793
      RHOW=EXP(-0.408*Z)                                                NUCMOD  1794
      GO TO 6                                                           NUCMOD  1795
    4 IF (Z.GT.15.0) GO TO 5                                            NUCMOD  1796
      RHOW=EXP(0.7403-0.5561*Z)                                         NUCMOD  1797
      GO TO 6                                                           NUCMOD  1798
    5 RHOW=0.0005                                                       NUCMOD  1799
    6 RETURN                                                            NUCMOD  1800
      END                                                               NUCMOD  1801
```

```
      SUBROUTINE WCO2P (ZF,ARG,IWR)                                     NUCMOD  1802
C                                                                       NUCMOD  1803
C     SUBROUTINE WCO2P COMPUTES CO2 CONCENTRATION.                      NUCMOD  1804
C     ZF IS ALTITUDE IN FEET.                                           NUCMOD  1805
C     ARG IS RETURNED VALUE.                                            NUCMOD  1806
C     IWR IS CODE, 0, RETURN DW/DZ IN ATMOS-CM/KM                       NUCMOD  1807
C                  1, RETURN W IN ATMOS-CM                              NUCMOD  1808
C                                                                       NUCMOD  1809
      DIMENSION A(4), B(4), ZC(3)                                       NUCMOD  1810
C                                                                       NUCMOD  1811
      DATA A/-0.1313,-0.3000,-0.1327,-0.1774/,B/5.520,7.460,5.285,6.112/ NUCMOD  1812
      DATA ZC/11.5,13.0,18.5/                                           NUCMOD  1813
C                                                                       NUCMOD  1814
      Z=ZF*0.0003048                                                    NUCMOD  1815
      DO 1 I=1,3                                                        NUCMOD  1816
      IF (Z.LE.ZC(I)) GO TO 2                                           NUCMOD  1817
    1 CONTINUE                                                          NUCMOD  1818
      I=4                                                               NUCMOD  1819
    2 ARG=EXP(A(I)*Z+B(I))                                              NUCMOD  1820
      IF (IWR.EQ.1) RETURN                                              NUCMOD  1821
      ARG=A(I)*ARG                                                      NUCMOD  1822
      RETURN                                                            NUCMOD  1823
      END                                                               NUCMOD  1824
```

APPENDIX B

QUANTA OUTPUT

```
PAGE   1

THERE ARE      1 MARVAL RUNS FOR THIS PROBLEM WITH      4 TARGETS,      3 SUB LOCATIONS,
            2 TYPES OF AIRCRAFT (TOTAL OF     20 ), AND      2 TYPES OF MISSILES.
 THE REQUIRED SIZE FOR THE INDX ARRAY IS        9
 THE SIZE YOU HAVE DESIGNATED IS        9

TARGET      VALUE  LOCATION (DEGREES)    RUNWAYS  AIRCRAFT
                                                    TYPE AND NUMBER

     1  1-.0000   13.5833  144.9167          1          1      2.0000
                                                        2      1.0000
                             TAKE-OFF SEQUENCE BY TYPE       1 1 2

     2   -.0000   44.8000   68.8333          1          1      1.0000
                                                        2      3.0000
                             TAKE-OFF SEQUENCE BY TYPE       1 2 2 2

     3    .0000   33.9167   84.5167          2          1      4.0000
                                                        2      5.0000
                             TAKE-OFF SEQUENCE BY TYPE       1 1 1 1 2 2 2 2 2

     4  1 .0000   33.4333  112.1667          1          1      4.0000
                                                        2      0.0000
                             TAKE-OFF SEQUENCE BY TYPE       1 1 1 1

 AIRCRAFT  RELATIVE      BRAKE  CENTROID
     TYPE     VALUE    RELEASE  DISTANCE         PSI       CAL
        1    2.5000     3.5000    3.5000      1.0000   96.0000
        2    1.0000     4.2500    5.5000      1.5000  120.0000

AIRCRAFT T-KE-OFF INTERVALS
      RUN AYS     TYPE1     TYPE2   MINUTES
            1         1         1      .2500
            1         1         2      .5000
            1         2         1      .5000
            1         2         2      .5000
            2         1         1      .1250
            2         1         2      .2500
            2         2         1      .2500
            2         2         2      .2500

 SUB LOCATIONS (DEGREES)        SUBS   MISSILES AND TYPE
   1   36.7500   74.0000          2          18        1
   2   26. 500   93.4500          3          12        2
   3   45.-500  126.0500          2          10        2

NUMBER OF SUBS OF TYPE    1 =    2

NUMBER OF SUBS OF TYPE    2 =    5

  MISSILE    LAUNCH                                MIN       MAX                HEIGHT       NUMBER
     TYPE  INTERVAL    RELIABILITY               RANGE     RANGE      YIELD    OF BURST   RVS/MISSILE
        1     .2000       .9000                250.000  2250.000  1000.0000    7000.00          2.0

  MISSILE    LAUNCH                                MIN       MAX                HEIGHT       NUMBER
     TYPE  INTERVAL    RELIABILITY               RANGE     RANGE      YIELD    OF BURST   RVS/MISSILE
        2     .2500       .8500                650.000  2000.000   800.0000    6000.00          3.0

MISSILE TY-E            TIME     RANGE
           1          3.5000  250.0000
                      5.5900  950.0000
```

```
                    7.5000 1600.0000
                    9.5000 2250.0000

 PAGE   2

MISSILE TYPE           TIME      RANGE
        2             5.0000  650.0000
                      7.5000 1100.0000
                     10.0000 1550.0000
                     12.5000 2000.0000

THE REQUIRED WORK2 ARRAY SIZE IS         656
THE PROVIDED WORK2 ARRAY SIZE IS         656
THE REQUIRED WORK1 ARRAY SIZE IS         806
THE PROVIDED WORK1 ARRAY SIZE IS         806
 THE REQUIRED SIZE FOR THE ALOC AND FLAMB ARRAYS IS      480
 THE SIZE YOU HAVE DESIGNATED IS      480
THE REQUIRED WORK2 ARRAY SIZE IS          72
THE PROVIDED WORK2 ARRAY SIZE IS         656
```

PAGE 3

SUBROUTINE PROCESS

AIRCRAFT TYPE 1 OF 2 TYPE(S)

VULNERABILITY CRITERIA FOR AIRCRAFT TYPE

1.00 PSI 96 CAL/CM**2

ATA INPUT FOR AIRCRAFT PROFILE

CARD NUMBER	GROUND RANGE	FLIGHT TIME	AIRCRAFT ALTITUDE	VELOCITY OF SOUND	MACH NUMBER	LEVEL-OFF ACCELERATION
1	0.00000000	0.00000000	0.00000000	1116.43553240	0.000	0.000
2	1500.00000000	7.50000000	5.00000000	1116.41737491	.130	0.000
3	3700.00000000	19.00000000	7.00000000	1116.40971183	.170	0.000
4	5700.00000000	30.00000000	16.00000000	1116.37522729	.200	0.000
5	9000.00000000	48.00000000	75.00000000	1116.14913559	.250	0.000
6	13000.00000000	63.00000000	230.00000000	1115.55494757	.340	0.000
7	20000.00000000	79.50000000	250.00000000	1115.47825509	.450	0.000
8	33000.00000000	105.00000000	500.00000000	1114.51915304	.600	0.000
9	50000.00000000	130.00000000	1250.00000000	1111.63866513	.750	0.000
10	63000.00000000	150.00000000	2300.00000000	1107.56910818	.800	0.000
11	72000.00000000	160.00000000	2800.00000000	1105.65638693	.900	0.000
12	90000.00000000	180.00000000	4500.00000000	1099.05971240	.900	0.000
13	98000.00000000	195.00000000	5800.00000000	1093.98835753	.900	0.000
14	105000.00000000	200.00000000	7000.00000000	1089.28615092	.900	0.000
15	110000.00000000	205.00000000	9200.00000000	1080.61229753	.900	0.000
16	115000.00000000	210.00000000	10000.00000000	1077.44085639	.900	0.000

INITIAL ALTITUDE OF MAXIMUM MACH 2800.0 FEET

ATA COMPUTED FOR AIRCRAFT TYPE 1 AT A TERMINAL ALTITUDE OF 10000.0 FEET

VELOCITY OF SOUND: 1077.44 FEET/SECOND

ACCELERATION COMPONENT: 0.000 FEET/SEC/SEC

MACH NUMBER: .900 TERMINAL .900

VELOCITY: 969.70 FEET/SEC FINAL 995.09 FEET/SEC

GROUND RANGE (FEET)	FLIGHT TIME (SEC)	GROUND RANGE (NM)	FLIGHT TIME (MIN)
0.	0.	0.	0.
.15000000E+04	.75000000E+01	.24671053E+00	.12500000E+00
.37000000E+04	.19000000E+02	.60855263E+00	.31666667E+00
.57000000E+04	.30000000E+02	.93750000E+00	.50000000E+00
.90000000E+04	.48000000E+02	.14802632E+01	.80000000E+00
.13000000E+05	.63000000E+02	.21381579E+01	.10500000E+01
.20000000E+05	.79500000E+02	.32894737E+01	.13250000E+01
.33000000E+05	.10500000E+03	.54276316E+01	.17500000E+01
.50000000E+05	.13000000E+03	.82236842E+01	.21666667E+01
.63000000E+05	.15000000E+03	.10361842E+02	.25000000E+01
.72000000E+05	.16000000E+03	.11842105E+02	.26666667E+01

.90000000E+05	.18000000E+03	.14802632E+02	.30000000E+01

.98000000E+05	.19500000E+03	.16118421E+02	.32500000E+01
.10500000E+06	.20000000E+03	.17269737E+02	.33333333E+01
.11000000E+06	.20500000E+03	.18092105E+02	.34166667E+01
.11500000E+06	.21000000E+03	.18914474E+02	.35000000E+01
.17318181E+06	.27000000E+03	.28483850E+02	.45000000E+01
.23136361E+06	.33000000E+03	.38053226E+02	.55000000E+01
.28954542E+06	.39000000E+03	.47622602E+02	.65000000E+01
.34772722E+06	.45000000E+03	.57191978E+02	.75000000E+01

SUBROUTINE PROCESS

AIRCRAFT TYPE 2 OF 2 TYPE(S)

VULNERABILITY CRITERIA FOR AIRCRAFT TYPE

1.50 PSI 120 CAL/CM**2

DATA INPUT FOR AIRCRAFT PROFILE

CARD NUMBER	GROUND RANGE	FLIGHT TIME	AIRCRAFT ALTITUDE	VELOCITY OF SOUND	MACH NUMBER	LEVEL-OFF ACCELERATION
1	0.00000000	0.00000000	0.00000000	1116.43653240	0.000	6.200
2	2000.00000000	20.00000000	9.00000000	1116.40204669	.140	6.200
3	3900.00000000	30.00000000	30.00000000	1116.32158255	.140	6.200
4	6200.00000000	40.00000000	92.00000000	1116.08398203	.190	6.200
5	9500.00000000	50.00000000	360.00000000	1115.05635218	.250	6.200
6	13400.00000000	60.00000000	500.00000000	1114.51915384	.310	6.200
7	15600.00000000	66.00000000	500.01000000	1114.51911546	.350	6.200
8	55900.00000000	125.00000000	500.11000000	1114.51873166	.680	6.200
9	61000.00000000	132.00000000	1425.00000000	1110.96327981	.700	6.200
10	68000.00000000	140.00000000	3950.00000000	1101.19625499	.720	6.200
11	75500.00000000	148.00000000	5000.00000000	1097.11196555	.730	6.200

INITIAL ALTITUDE OF MAXIMUM MACH 7600.0 FEET

DATA COMPUTED FOR AIRCRAFT TYPE 2 AT A TERMINAL ALTITUDE OF 5000.0 FEET

VELOCITY OF SOUND; 1097.11 FEET/SECOND

ACCELERATION COMPONENT; 6.200 FEET/SEC/SEC

MACH NUMBER; .730 TERMINAL .849

VELOCITY; 800.89 FEET/SEC FINAL 922.80 FEET/SEC

GROUND RANGE (FEET)	FLIGHT TIME (SEC)	GROUND RANGE (NM)	FLIGHT TIME (MIN)
0.	0.	0.	0.
.20000000E+04	.20000000E+02	.32894737E+00	.33333333E+00
.39000000E+04	.30000000E+02	.64144737E+00	.50000000E+00
.62000000E+04	.40000000E+02	.10197368E+01	.66666667E+00
.95000000E+04	.50000000E+02	.15625000E+01	.83333333E+00
.13400000E+05	.60000000E+02	.22039474E+01	.10000000E+01
.15600000E+05	.66000000E+02	.25657895E+01	.11000000E+01
.55900000E+05	.12500000E+03	.91940789E+01	.20833333E+01
.61000000E+05	.13200000E+03	.10032895E+02	.22000000E+01
.68000000E+05	.14000000E+03	.11184211E+02	.23333333E+01
.75500000E+05	.14800000E+03	.12417763E+02	.24666667E+01
.77086767E+05	.14996628E+03	.12678744E+02	.24994381E+01
.78697504E+05	.15193257E+03	.12943666E+02	.25322095E+01
.80332212E+05	.15369885E+03	.13212535E+02	.25649809E+01
.81990892E+05	.15566514E+03	.13485344E+02	.25977523E+01
.83673542E+05	.15763142E+03	.13762096E+02	.26305237E+01

.85380163E+05	.15979771E+03	.14042790E+02	.26632951E+01

PAGE 6

.87110755E+05	.16176399E+03	.14327427E+02	.26960665E+01
.92446356E+05	.16766285E+03	.15204993E+02	.27943808E+01
.14781444E+06	.22766285E+03	.24311585E+02	.37943808E+01

THERE WILL BE 1 MARVAL RUN(S) FOR THIS CASE.
THE CUT-OFF VALUES ARE .50000

```
********** AIRCRAFT TYPE    1 .VS. MISSILE TYPE    1**********

WHEN DETONATION IS    0.0 NM FROM CENTROID, LETHAL AREA =   177.43 SQUARE NM
WHEN DETONATION IS    1.0 NM FROM CENTROID, LETHAL AREA =   149.20 SQUARE NM
WHEN DETONATION IS    2.0 NM FROM CENTROID, LETHAL AREA =   126.61 SQUARE NM
WHEN DETONATION IS    3.0 NM FROM CENTROID, LETHAL AREA =   111.74 SQUARE NM
WHEN DETONATION IS    4.0 NM FROM CENTROID, LETHAL AREA =   103.06 SQUARE NM
WHEN DETONATION IS    5.0 NM FROM CENTROID, LETHAL AREA =   100.15 SQUARE NM
WHEN DETONATION IS    6.0 NM FROM CENTROID, LETHAL AREA =   102.17 SQUARE NM
WHEN DETONATION IS    7.0 NM FROM CENTROID, LETHAL AREA =   106.15 SQUARE NM
WHEN DETONATION IS    8.0 NM FROM CENTROID, LETHAL AREA =   108.75 SQUARE NM
WHEN DETONATION IS    9.0 NM FROM CENTROID, LETHAL AREA =   112.26 SQUARE NM
WHEN DETONATION IS   10.0 NM FROM CENTROID, LETHAL AREA =   114.96 SQUARE NM
WHEN DETONATION IS   11.0 NM FROM CENTROID, LETHAL AREA =   119.49 SQUARE NM
WHEN DETONATION IS   12.0 NM FROM CENTROID, LETHAL AREA =   129.11 SQUARE NM
WHEN DETONATION IS   15.0 NM FROM CENTROID, LETHAL AREA =   133.85 SQUARE NM
WHEN DETONATION IS   20.0 NM FROM CENTROID, LETHAL AREA =   128.41 SQUARE NM
WHEN DETONATION IS   25.0 NM FROM CENTROID, LETHAL AREA =   135.33 SQUARE NM
WHEN DETONATION IS   30.0 NM FROM CENTROID, LETHAL AREA =   139.66 SQUARE NM
WHEN DETONATION IS   35.0 NM FROM CENTROID, LETHAL AREA =   142.79 SQUARE NM
WHEN DETONATION IS   40.0 NM FROM CENTROID, LETHAL AREA =   145.13 SQUARE NM
WHEN DETONATION IS   45.0 NM FROM CENTROID, LETHAL AREA =   146.94 SQUARE NM
WHEN DETONATION IS   50.0 NM FROM CENTROID, LETHAL AREA =   148.39 SQUARE NM
WHEN DETONATION IS   55.0 NM FROM CENTROID, LETHAL AREA =   149.57 SQUARE NM
WHEN DETONATION IS   60.0 NM FROM CENTROID, LETHAL AREA =   150.56 SQUARE NM
WHEN DETONATION IS   65.0 NM FROM CENTROID, LETHAL AREA =   151.39 SQUARE NM
WHEN DETONATION IS   70.0 NM FROM CENTROID, LETHAL AREA =   152.10 SQUARE NM
WHEN DETONATION IS   75.0 NM FROM CENTROID, LETHAL AREA =   152.72 SQUARE NM
WHEN DETONATION IS   80.0 NM FROM CENTROID, LETHAL AREA =   153.26 SQUARE NM
WHEN DETONATION IS   85.0 NM FROM CENTROID, LETHAL AREA =   153.74 SQUARE NM
WHEN DETONATION IS   90.0 NM FROM CENTROID, LETHAL AREA =   154.16 SQUARE NM
WHEN DETONATION IS   95.0 NM FROM CENTROID, LETHAL AREA =   154.54 SQUARE NM


DETONATION IS    0.0 NM FROM BRAKE RELEASE, KILL LINE IS   0.0 PLUS   4.7 NM
DETONATION IS    1.0 NM FROM BRAKE RELEASE, KILL LINE IS   1.0 PLUS   4.4 NM
DETONATION IS    2.0 NM FROM BRAKE RELEASE, KILL LINE IS   2.0 PLUS   4.1 NM
DETONATION IS    3.0 NM FROM BRAKE RELEASE, KILL LINE IS   3.0 PLUS   3.8 NM
DETONATION IS    4.0 NM FROM BRAKE RELEASE, KILL LINE IS   4.0 PLUS   3.5 NM
DETONATION IS    5.0 NM FROM BRAKE RELEASE, KILL LINE IS   5.0 PLUS   3.3 NM
DETONATION IS    6.0 NM FROM BRAKE RELEASE, KILL LINE IS   6.0 PLUS   3.3 NM
DETONATION IS    7.0 NM FROM BRAKE RELEASE, KILL LINE IS   7.0 PLUS   3.5 NM
DETONATION IS    8.0 NM FROM BRAKE RELEASE, KILL LINE IS   8.0 PLUS   3.1 NM
DETONATION IS    9.0 NM FROM BRAKE RELEASE, KILL LINE IS   9.0 PLUS   2.5 NM
DETONATION IS   10.0 NM FROM BRAKE RELEASE, KILL LINE IS  10.0 PLUS   2.9 NM
DETONATION IS   11.0 NM FROM BRAKE RELEASE, KILL LINE IS  11.0 PLUS   2.7 NM
DETONATION IS   12.0 NM FROM BRAKE RELEASE, KILL LINE IS  11.6 PLUS   2.6 NM
```

```
********** AIRCRAFT TYPE    1 .VS. MISSILE TYPE    2**********

WHEN DETONATION IS    0.0 NM FROM CENTROID, LETHAL AREA =   142.39 SQUARE NM
WHEN DETONATION IS    1.0 NM FROM CENTROID, LETHAL AREA =   118.97 SQUARE NM
WHEN DETONATION IS    2.0 NM FROM CENTROID, LETHAL AREA =   101.54 SQUARE NM
WHEN DETONATION IS    3.0 NM FROM CENTROID, LETHAL AREA =    91.57 SQUARE NM
WHEN DETONATION IS    4.0 NM FROM CENTROID, LETHAL AREA =    85.60 SQUARE NM
WHEN DETONATION IS    5.0 NM FROM CENTROID, LETHAL AREA =    84.55 SQUARE NM
WHEN DETONATION IS    6.0 NM FROM CENTROID, LETHAL AREA =    88.04 SQUARE NM
WHEN DETONATION IS    7.0 NM FROM CENTROID, LETHAL AREA =    93.45 SQUARE NM
WHEN DETONATION IS    8.0 NM FROM CENTROID, LETHAL AREA =    97.02 SQUARE NM
WHEN DETONATION IS    9.0 NM FROM CENTROID, LETHAL AREA =   100.59 SQUARE NM
WHEN DETONATION IS   10.0 NM FROM CENTROID, LETHAL AREA =   102.75 SQUARE NM
WHEN DETONATION IS   11.0 NM FROM CENTROID, LETHAL AREA =   105.90 SQUARE NM
WHEN DETONATION IS   12.0 NM FROM CENTROID, LETHAL AREA =   115.19 SQUARE NM
WHEN DETONATION IS   15.0 NM FROM CENTROID, LETHAL AREA =   119.66 SQUARE NM
WHEN DETONATION IS   20.0 NM FROM CENTROID, LETHAL AREA =   118.19 SQUARE NM
WHEN DETONATION IS   25.0 NM FROM CENTROID, LETHAL AREA =   121.85 SQUARE NM
WHEN DETONATION IS   30.0 NM FROM CENTROID, LETHAL AREA =   125.41 SQUARE NM
WHEN DETONATION IS   35.0 NM FROM CENTROID, LETHAL AREA =   127.99 SQUARE NM
WHEN DETONATION IS   40.0 NM FROM CENTROID, LETHAL AREA =   129.92 SQUARE NM
WHEN DETONATION IS   45.0 NM FROM CENTROID, LETHAL AREA =   131.41 SQUARE NM
WHEN DETONATION IS   50.0 NM FROM CENTROID, LETHAL AREA =   132.61 SQUARE NM
WHEN DETONATION IS   55.0 NM FROM CENTROID, LETHAL AREA =   133.58 SQUARE NM
WHEN DETONATION IS   60.0 NM FROM CENTROID, LETHAL AREA =   134.39 SQUARE NM
WHEN DETONATION IS   65.0 NM FROM CENTROID, LETHAL AREA =   135.08 SQUARE NM
WHEN DETONATION IS   70.0 NM FROM CENTROID, LETHAL AREA =   135.67 SQUARE NM
WHEN DETONATION IS   75.0 NM FROM CENTROID, LETHAL AREA =   136.18 SQUARE NM
WHEN DETONATION IS   80.0 NM FROM CENTROID, LETHAL AREA =   136.62 SQUARE NM
WHEN DETONATION IS   85.0 NM FROM CENTROID, LETHAL AREA =   137.01 SQUARE NM
WHEN DETONATION IS   90.0 NM FROM CENTROID, LETHAL AREA =   137.36 SQUARE NM
WHEN DETONATION IS   95.0 NM FROM CENTROID, LETHAL AREA =   137.68 SQUARE NM


DETONATION IS    0.0 NM FROM BRAKE RELEASE, KILL LINE IS   0.0 PLUS   4.4 NM
DETONATION IS    1.0 NM FROM BRAKE RELEASE, KILL LINE IS   1.0 PLUS   4.2 NM
DETONATION IS    2.0 NM FROM BRAKE RELEASE, KILL LINE IS   2.0 PLUS   4.0 NM
DETONATION IS    3.0 NM FROM BRAKE RELEASE, KILL LINE IS   3.0 PLUS   3.7 NM
DETONATION IS    4.0 NM FROM BRAKE RELEASE, KILL LINE IS   4.0 PLUS   3.4 NM
DETONATION IS    5.0 NM FROM BRAKE RELEASE, KILL LINE IS   5.0 PLUS   3.1 NM
DETONATION IS    6.0 NM FROM BRAKE RELEASE, KILL LINE IS   6.0 PLUS   2.9 NM
DETONATION IS    7.0 NM FROM BRAKE RELEASE, KILL LINE IS   7.0 PLUS   2.9 NM
DETONATION IS    8.0 NM FROM BRAKE RELEASE, KILL LINE IS   8.0 PLUS   3.2 NM
DETONATION IS    9.0 NM FROM BRAKE RELEASE, KILL LINE IS   9.0 PLUS   2.8 NM
DETONATION IS   10.0 NM FROM BRAKE RELEASE, KILL LINE IS  10.0 PLUS   2.4 NM
DETONATION IS   11.0 NM FROM BRAKE RELEASE, KILL LINE IS  10.5 PLUS   2.7 NM
DETONATION IS   12.0 NM FROM BRAKE RELEASE, KILL LINE IS  11.0 PLUS   2.4 NM
```

********** AIRCRAFT TYPE 2 .VS. MISSILE TYPE 1**********

WHEN DETONATION IS 0.0 NM FROM CENTROID, LETHAL AREA = 59.54 SQUARE NM
WHEN DETONATION IS 1.0 NM FROM CENTROID, LETHAL AREA = 48.39 SQUARE NM
WHEN DETONATION IS 2.0 NM FROM CENTROID, LETHAL AREA = 44.55 SQUARE NM
WHEN DETONATION IS 3.0 NM FROM CENTROID, LETHAL AREA = 45.22 SQUARE NM
WHEN DETONATION IS 4.0 NM FROM CENTROID, LETHAL AREA = 50.53 SQUARE NM
WHEN DETONATION IS 5.0 NM FROM CENTROID, LETHAL AREA = 54.52 SQUARE NM
WHEN DETONATION IS 6.0 NM FROM CENTROID, LETHAL AREA = 56.61 SQUARE NM
WHEN DETONATION IS 7.0 NM FROM CENTROID, LETHAL AREA = 60.22 SQUARE NM
WHEN DETONATION IS 8.0 NM FROM CENTROID, LETHAL AREA = 66.12 SQUARE NM
WHEN DETONATION IS 9.0 NM FROM CENTROID, LETHAL AREA = 71.07 SQUARE NM
WHEN DETONATION IS 10.0 NM FROM CENTROID, LETHAL AREA = 77.38 SQUARE NM
WHEN DETONATION IS 11.0 NM FROM CENTROID, LETHAL AREA = 84.67 SQUARE NM
WHEN DETONATION IS 12.0 NM FROM CENTROID, LETHAL AREA = 91.23 SQUARE NM
WHEN DETONATION IS 15.0 NM FROM CENTROID, LETHAL AREA = 104.46 SQUARE NM
WHEN DETONATION IS 20.0 NM FROM CENTROID, LETHAL AREA = 118.72 SQUARE NM
WHEN DETONATION IS 25.0 NM FROM CENTROID, LETHAL AREA = 127.13 SQUARE NM
WHEN DETONATION IS 30.0 NM FROM CENTROID, LETHAL AREA = 132.62 SQUARE NM
WHEN DETONATION IS 35.0 NM FROM CENTROID, LETHAL AREA = 136.52 SQUARE NM
WHEN DETONATION IS 40.0 NM FROM CENTROID, LETHAL AREA = 139.42 SQUARE NM
WHEN DETONATION IS 45.0 NM FROM CENTROID, LETHAL AREA = 141.68 SQUARE NM
WHEN DETONATION IS 50.0 NM FROM CENTROID, LETHAL AREA = 143.48 SQUARE NM
WHEN DETONATION IS 55.0 NM FROM CENTROID, LETHAL AREA = 144.95 SQUARE NM
WHEN DETONATION IS 60.0 NM FROM CENTROID, LETHAL AREA = 146.17 SQUARE NM
WHEN DETONATION IS 65.0 NM FROM CENTROID, LETHAL AREA = 147.21 SQUARE NM
WHEN DETONATION IS 70.0 NM FROM CENTROID, LETHAL AREA = 148.09 SQUARE NM
WHEN DETONATION IS 75.0 NM FROM CENTROID, LETHAL AREA = 148.86 SQUARE NM
WHEN DETONATION IS 80.0 NM FROM CENTROID, LETHAL AREA = 149.53 SQUARE NM
WHEN DETONATION IS 85.0 NM FROM CENTROID, LETHAL AREA = 150.12 SQUARE NM
WHEN DETONATION IS 90.0 NM FROM CENTROID, LETHAL AREA = 150.65 SQUARE NM
WHEN DETONATION IS 95.0 NM FROM CENTROID, LETHAL AREA = 151.12 SQUARE NM

DETONATION IS 0.0 NM FROM BRAKE RELEASE, KILL LINE IS 0.0 PLUS 3.5 NM
DETONATION IS 1.0 NM FROM BRAKE RELEASE, KILL LINE IS 1.0 PLUS 3.4 NM
DETONATION IS 2.0 NM FROM BRAKE RELEASE, KILL LINE IS 2.0 PLUS 3.3 NM
DETONATION IS 3.0 NM FROM BRAKE RELEASE, KILL LINE IS 3.0 PLUS 3.2 NM
DETONATION IS 4.0 NM FROM BRAKE RELEASE, KILL LINE IS 4.0 PLUS 3.0 NM
DETONATION IS 5.0 NM FROM BRAKE RELEASE, KILL LINE IS 5.0 PLUS 2.7 NM
DETONATION IS 6.0 NM FROM BRAKE RELEASE, KILL LINE IS 6.0 PLUS 2.6 NM
DETONATION IS 7.0 NM FROM BRAKE RELEASE, KILL LINE IS 7.0 PLUS 2.4 NM
DETONATION IS 8.0 NM FROM BRAKE RELEASE, KILL LINE IS 8.0 PLUS 2.3 NM
DETONATION IS 9.0 NM FROM BRAKE RELEASE, KILL LINE IS 8.9 PLUS 2.4 NM
DETONATION IS 10.0 NM FROM BRAKE RELEASE, KILL LINE IS 9.5 PLUS 2.4 NM
DETONATION IS 11.0 NM FROM BRAKE RELEASE, KILL LINE IS 10.1 PLUS 2.4 NM
DETONATION IS 12.0 NM FROM BRAKE RELEASE, KILL LINE IS 10.5 PLUS 2.4 NM

```
********** AIRCRAFT TYPE     2 .VS. MISSILE TYPE     2**********

WHEN DETONATION IS    0.0 NM FROM CENTROID, LETHAL AREA =    46.51 SQUARE NM
WHEN DETONATION IS    1.0 NM FROM CENTROID, LETHAL AREA =    39.71 SQUARE NM
WHEN DETONATION IS    2.0 NM FROM CENTROID, LETHAL AREA =    37.25 SQUARE NM
WHEN DETONATION IS    3.0 NM FROM CENTROID, LETHAL AREA =    38.88 SQUARE NM
WHEN DETONATION IS    4.0 NM FROM CENTROID, LETHAL AREA =    44.55 SQUARE NM
WHEN DETONATION IS    5.0 NM FROM CENTROID, LETHAL AREA =    48.81 SQUARE NM
WHEN DETONATION IS    6.0 NM FROM CENTROID, LETHAL AREA =    50.80 SQUARE NM
WHEN DETONATION IS    7.0 NM FROM CENTROID, LETHAL AREA =    54.09 SQUARE NM
WHEN DETONATION IS    8.0 NM FROM CENTROID, LETHAL AREA =    59.27 SQUARE NM
WHEN DETONATION IS    9.0 NM FROM CENTROID, LETHAL AREA =    64.19 SQUARE NM
WHEN DETONATION IS   10.0 NM FROM CENTROID, LETHAL AREA =    70.43 SQUARE NM
WHEN DETONATION IS   11.0 NM FROM CENTROID, LETHAL AREA =    76.80 SQUARE NM
WHEN DETONATION IS   12.0 NM FROM CENTROID, LETHAL AREA =    81.02 SQUARE NM
WHEN DETONATION IS   15.0 NM FROM CENTROID, LETHAL AREA =    91.02 SQUARE NM
WHEN DETONATION IS   20.0 NM FROM CENTROID, LETHAL AREA =   101.59 SQUARE NM
WHEN DETONATION IS   25.0 NM FROM CENTROID, LETHAL AREA =   107.84 SQUARE NM
WHEN DETONATION IS   30.0 NM FROM CENTROID, LETHAL AREA =   111.94 SQUARE NM
WHEN DETONATION IS   35.0 NM FROM CENTROID, LETHAL AREA =   114.85 SQUARE NM
WHEN DETONATION IS   40.0 NM FROM CENTROID, LETHAL AREA =   117.02 SQUARE NM
WHEN DETONATION IS   45.0 NM FROM CENTROID, LETHAL AREA =   118.71 SQUARE NM
WHEN DETONATION IS   50.0 NM FROM CENTROID, LETHAL AREA =   120.06 SQUARE NM
WHEN DETONATION IS   55.0 NM FROM CENTROID, LETHAL AREA =   121.16 SQUARE NM
WHEN DETONATION IS   60.0 NM FROM CENTROID, LETHAL AREA =   122.07 SQUARE NM
WHEN DETONATION IS   65.0 NM FROM CENTROID, LETHAL AREA =   122.85 SQUARE NM
WHEN DETONATION IS   70.0 NM FROM CENTROID, LETHAL AREA =   123.51 SQUARE NM
WHEN DETONATION IS   75.0 NM FROM CENTROID, LETHAL AREA =   124.08 SQUARE NM
WHEN DETONATION IS   80.0 NM FROM CENTROID, LETHAL AREA =   124.59 SQUARE NM
WHEN DETONATION IS   85.0 NM FROM CENTROID, LETHAL AREA =   125.03 SQUARE NM
WHEN DETONATION IS   90.0 NM FROM CENTROID, LETHAL AREA =   125.42 SQUARE NM
WHEN DETONATION IS   95.0 NM FROM CENTROID, LETHAL AREA =   125.78 SQUARE NM


DETONATION IS    0.0 NM FROM BRAKE RELEASE, KILL LINE IS   0.0 PLUS   3.3 NM
DETONATION IS    1.0 NM FROM BRAKE RELEASE, KILL LINE IS   1.0 PLUS   3.1 NM
DETONATION IS    2.0 NM FROM BRAKE RELEASE, KILL LINE IS   2.0 PLUS   3.0 NM
DETONATION IS    3.0 NM FROM BRAKE RELEASE, KILL LINE IS   3.0 PLUS   3.0 NM
DETONATION IS    4.0 NM FROM BRAKE RELEASE, KILL LINE IS   4.0 PLUS   2.8 NM
DETONATION IS    5.0 NM FROM BRAKE RELEASE, KILL LINE IS   5.0 PLUS   2.6 NM
DETONATION IS    6.0 NM FROM BRAKE RELEASE, KILL LINE IS   6.0 PLUS   2.4 NM
DETONATION IS    7.0 NM FROM BRAKE RELEASE, KILL LINE IS   7.0 PLUS   2.3 NM
DETONATION IS    8.0 NM FROM BRAKE RELEASE, KILL LINE IS   7.9 PLUS   2.1 NM
DETONATION IS    9.0 NM FROM BRAKE RELEASE, KILL LINE IS   8.6 PLUS   2.1 NM
DETONATION IS   10.0 NM FROM BRAKE RELEASE, KILL LINE IS   9.1 PLUS   2.2 NM
DETONATION IS   11.0 NM FROM BRAKE RELEASE, KILL LINE IS   9.5 PLUS   2.2 NM
DETONATION IS   12.0 NM FROM BRAKE RELEASE, KILL LINE IS   9.9 PLUS   2.2 NM
```

```
THE DISTANCE TO TARGET   1 FROM SUB  1 IS  4006.5347      THE AVERAGE DISTANCE TO THE TURN POINT IS   4.17 NM

THE DISTANCE TO TARGET   1 FROM SUB  2 IS  2994.5799      THE AVERAGE DISTANCE TO THE TURN POINT IS   4.17 NM

THE DISTANCE TO TARGET   1 FROM SUB  3 IS  2151.6043      THE AVERAGE DISTANCE TO THE TURN POINT IS   4.17 NM

THE DISTANCE TO TARGET   2 FROM SUB  1 IS   537.4366      THE AVERAGE DISTANCE TO THE TURN POINT IS   5.00 NM
AFTER   4.421247 MINUTES, WHEN WEAPON    1 ARRIVES, THE RANGES(NM) OF AIRCRAFT ARE
     1.5274     .3120    0.0000    0.0000

THE DISTANCE TO TARGET   2 FROM SUB  2 IS  1547.3286      THE AVERAGE DISTANCE TO THE TURN POINT IS   5.00 NM
AFTER   9.435159 MINUTES, WHEN WEAPON   19 ARRIVES, THE RANGES(NM) OF AIRCRAFT ARE
    47.4654   44.2621   39.7088   35.1555

THE DISTANCE TO TARGET   2 FROM SUB  3 IS  2360.6136      THE AVERAGE DISTANCE TO THE TURN POINT IS   5.00 NM

THE DISTANCE TO TARGET   3 FROM SUB  1 IS   542.4627      THE AVERAGE DISTANCE TO THE TURN POINT IS   4.61 NM
AFTER   4.435608 MINUTES, WHEN WEAPON    1 ARRIVES, THE RANGES(NM) OF AIRCRAFT ARE
     1.5594    1.3063    1.0649     .8659     .1661    0.0000    0.0000    0.0000    0.0000

THE DISTANCE TO TARGET   3 FROM SUB  2 IS   579.1663      THE AVERAGE DISTANCE TO THE TURN POINT IS   4.61 NM

THE DISTANCE TO TARGET   3 FROM SUB  3 IS  2016.5589      THE AVERAGE DISTANCE TO THE TURN POINT IS   4.61 NM

THE DISTANCE TO TARGET   4 FROM SUB  1 IS  1874.3571      THE AVERAGE DISTANCE TO THE TURN POINT IS   3.50 NM
AFTER   9.444176 MINUTES, WHEN WEAPON    1 ARRIVES, THE RANGES(NM) OF AIRCRAFT ARE
    31.7772   29.3851   25.9958   24.5150

THE DISTANCE TO TARGET   4 FROM SUB  2 IS  1017.3580      THE AVERAGE DISTANCE TO THE TURN POINT IS   3.50 NM
AFTER   7.40878 MINUTES, WHEN WEAPON   19 ARRIVES, THE RANGES(NM) OF AIRCRAFT ARE
    19.3114   16.5580   15.0455   12.9978

THE DISTANCE TO TARGET   4 FROM SUB  3 IS   973.0329      THE AVERAGE DISTANCE TO THE TURN POINT IS   3.50 NM
AFTER   5.494627 MINUTES, WHEN WEAPON   31 ARRIVES, THE RANGES(NM) OF AIRCRAFT ARE
    16.7157   15.0540   13.0334   10.7267
```

TARGET NUMBER	NUMBER OF WEAPONS ALLOCATED TO THE BASE	RUNWAY	TURN PT	BASE VALUE	PROB. OF KILL	AIRCRAFT TYPE	AIRCRAFT VALUE	KILL VALUE	PROB. OF KILL
1	0.0	0.0	15.0	15.0000	0.0000				
						1	2.5000	0.0000	0.0000
						1	2.5000	0.0000	0.0000
						2	1.0000	0.0000	0.0000
		TARGET 1 HAS A TOTAL KILL OF 0.0000							
2	0.0	0.0	15.0	2.0000	0.0000				
						1	2.5000	.5093	.2757
						2	1.0000	.2471	.2471
						2	1.0000	.2471	.2471
						2	1.0000	.2471	.2471
		TARGET 2 HAS A TOTAL KILL OF 1.4355							
3	0.0	0.0	15.0	0.0000	0.0000				
						1	2.5000	0.0000	0.0000
						1	2.5000	0.0000	0.0000
						1	2.5000	0.0000	0.0000
						1	2.5000	0.0000	0.0000
						2	1.0000	0.0000	0.0000
						2	1.0000	0.0000	0.0000
						2	1.0000	0.0000	0.0000
						2	1.0000	0.0000	0.0000
						2	1.0000	0.0000	0.0000
		TARGET 3 HAS A TOTAL KILL OF 0.0000							
4	0.0	0.0	14.0	10.0000	0.0000				
						1	2.5000	2.1411	.8564
						1	2.5000	2.1411	.8564
						1	2.5000	2.1411	.8564
						1	2.5000	2.1411	.8564
		TARGET 4 HAS A TOTAL KILL OF 8.5644							
******0.000 TOTALS **********			59.00						
						1	27.5000	9.2537	3.7013
						2	9.0000	.7412	.7412

EXPECTED VALUE KILLED = 9.9949

NUMBER OF WEAPONS NOT ALLOCATED = 25

```
ITERATION   54
THE MULTIPLIER MATRIX HAS CONVERGED USING EPSILON =      .1000  THE MAXIMUM DELTA LAMBDA FOUND:   .092269

ITERATION   57
THE MULTIPLIER MATRIX HAS CONVERGED USING EPSILON =      .0100  THE MAXIMUM DELTA LAMBDA FOUND:   .006637

ITERATION   67
MULTIPLIER MATRIX CONVERGED WITHIN TOLERANCE OF     .010000
```

```
*****************************PRINTOUT OF THE OPTIMUM ALLOCATION OF MISSILES**************************

TARGET  1   .VS. SUB LOCATION        BASE    RUNWAY   TURN PT

TARGET  2   .VS. SUB LOCATION        BASE    RUNWAY   TURN PT
                 1 ( 2 SUBS)          1.0      1.0       0.0 FROM SALVO  4 (WEAPON GROUP   4 )
                 1 ( 2 SUBS)          0.0      1.0       0.0 FROM SALVO  7 (WEAPON GROUP   7 )

TARGET  3   .VS. SUB LOCATION        BASE    RUNWAY   TURN PT
                 1 ( 2 SUBS)          0.0      1.0       0.0 FROM SALVO  2 (WEAPON GROUP   2 )
                 1 ( 2 SUBS)          0.0      1.0       0.0 FROM SALVO  5 (WEAPON GROUP   5 )
                 1 ( 2 SUBS)          0.0      1.0       0.0 FROM SALVO  6 (WEAPON GROUP   6 )
                 1 ( 2 SUBS)          0.0      1.0       0.0 FROM SALVO  7 (WEAPON GROUP   7 )

TARGET  4   .VS. SUB LOCATION        BASE    RUNWAY   TURN PT
                 1 ( 2 SUBS)          1.0      0.0       0.0 FROM SALVO  1 (WEAPON GROUP   1 )
                 1 ( 2 SUBS)          1.0      0.0       0.0 FROM SALVO  2 (WEAPON GROUP   2 )
                 2 ( 3 SUBS)          0.0      0.0       2.0 FROM SALVO  1 (WEAPON GROUP  19 )
                 3 ( 2 SUBS)          0.0      0.0       2.0 FROM SALVO  1 (WEAPON GROUP  31 )

        SUB LOCATION   1 ( 2 SUBS)
                1.0 MISSILES FROM SALVO  1 TO TARGET   4 (  1.0 TO BASE)
                1.0 MISSILES FROM SALVO  2 TO TARGET   3 (  0.0 TO BASE)
                1.0 MISSILES FROM SALVO  2 TO TARGET   4 (  1.0 TO BASE)
                2.0 MISSILES FROM SALVO  4 TO TARGET   2 (  1.0 TO BASE)
                1.0 MISSILES FROM SALVO  5 TO TARGET   3 (  0.0 TO BASE)
                1.0 MISSILES FROM SALVO  6 TO TARGET   3 (  0.0 TO BASE)
                1.0 MISSILES FROM SALVO  7 TO TARGET   2 (  0.0 TO BASE)
                1.0 MISSILES FROM SALVO  7 TO TARGET   3 (  0.0 TO BASE)

        SUB LOCATION   2 ( 3 SUBS)
                2.0 MISSILES FROM SALVO  1 TO TARGET   4 (  0.0 TO BASE)

        SUB LOCATION   3 ( 2 SUBS)
                2.0 MISSILES FROM SALVO  1 TO TARGET   4 (  0.0 TO BASE)
```

THIS IS THE NUMBER 1 MARVAL RUN. CUT = .50000

TARGET NUMBER	NUMBER OF WEAPONS ALLOCATED TO THE BASE	RUNWAY	TURN PT	BASE VALUE	PROB. OF KILL	AIRCRAFT TYPE	AIRCRAFT VALUE	KILL VALUE	PROB. OF KILL
1	0.0	0.0	0.0	15.0000	0.0000				
						1	2.5000	0.0000	0.0000
						1	2.5000	0.0000	0.0000
						2	1.0000	0.0000	0.0000
		TARGET 1 HAS A TOTAL KILL OF			0.0000				
2	1.0	2.0	0.0	2.0000	.9600				
						1	2.5000	2.2829	.9132
						2	1.0000	.8455	.8455
						2	1.0000	.8455	.8455
						2	1.0000	.8455	.8455
		TARGET 2 HAS A TOTAL KILL OF			6.6194				
3	0.0	4.0	0.0	0.0000	0.0000				
						1	2.5000	2.4868	.9947
						1	2.5000	2.4868	.9947
						1	2.5000	2.4868	.9947
						1	2.5000	2.4868	.9947
						2	1.0000	.9776	.9776
						2	1.0000	.9776	.9776
						2	1.0000	.9776	.9776
						2	1.0000	.9776	.9776
						2	1.0000	.9776	.9776
		TARGET 3 HAS A TOTAL KILL OF			14.8351				
4	2.0	0.0	4.0	10.0000	.9900				
						1	2.5000	2.2630	.9052
						1	2.5000	2.2630	.9052
						1	2.5000	2.2630	.9052
						1	2.5000	2.2630	.9052
		TARGET 4 HAS A TOTAL KILL OF			18.9522				
********** TOTALS **********			13.00						
						1	27.5000	21.2821	8.5128
						2	9.0000	7.4246	7.4245

EXPECTED VALUE KILLED = 40.4066

NUMBER OF WEAPONS NOT ALLOCATED = 71